

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number  
**WO 02/087218 A2**

(51) International Patent Classification<sup>7</sup>: **H04N**

(21) International Application Number: **PCT/US02/13004**

(22) International Filing Date: 22 April 2002 (22.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/285,201 20 April 2001 (20.04.2001) US

(71) Applicant (for all designated States except US):  
**SARNOFF CORPORATION** [US/US]; 201 Washington Road, Princeton, NJ 08543-5300 (US).

(71) Applicant and

(72) Inventor (for all designated States except US): **KEWAZINGA CORP.** [US/US]; 191 Post Road West, Westport, CT 06880 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SOROKIN, Scott**

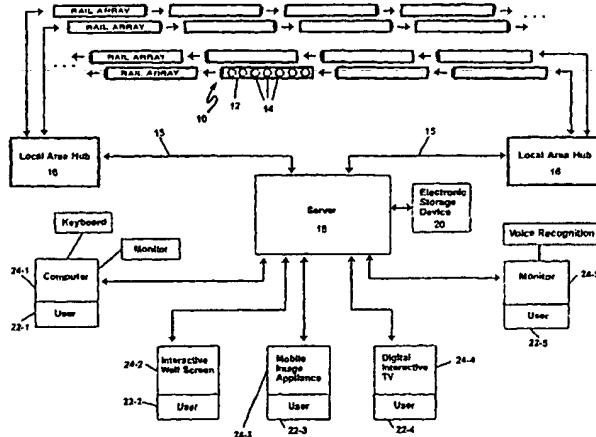
[US/US]; 75 West End Avenue, Suite P30A, New York, NY 10023 (US). **WEBER, Andrew, H.** [US/US]; 344 W. 72 Street, #108, New York, NY 10023 (US). **WORLEY, David, C.** [US/US]; 2 Kirsten Place, Weston, CT 06883 (US). **GENDEL, Gary** [US/US]; 17 Dogwood Drive, Hillsborough, NJ 08844 (US). **KEITH, J., Hanna** [GB/US]; 212 John Street, Princeton, NJ 08542 (US). **RAKESH, Kumar** [IN/US]; 966 Ridge Road, Princeton, NJ 08540 (US). **SUPUN, Samarasakera** [LK/US]; 24 Blackstone Drive, Princeton, NJ 08540 (US).

(74) Agents: **DIBERNARDO, Ian, G.** et al.; Stroock & Stroock & Lavan LLP, 180 Maiden Lane, New York, NY 10038 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: NAVIGABLE CAMERA ARRAY AND VIEWER THEREFORE



WO 02/087218 A2

(57) Abstract: User interface for the controlling of time and viewpoint, and warping of imagery in response to user inputs. The input to the interface can be either a raw set of pre-synthesized image data, or a set of original image data together with a set of flow-fields. In both cases the interface allows the user to navigate the data both in space and in time, for example, with the use of two slider controls, a single graphical control (e.g., a four-quadrant button, the quadrants corresponding to left, right, forward in time, backward in time), and the like. In one embodiment, the interface thus works generally according to the following four steps. Before being read into the interface, the data is first organized. The second step generally involves mapping the desired view given by the interface, and the next likely desired-view, onto the source data. The third step uses the mapping performed in the second step to fetch the data for the desired view and the next (anticipated) desired view into local memory. The forth step involves processing the data for the desired view as required, displaying the view on the screen, and continuing to fetch data for the next desired view.



(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— without international search report and to be republished upon receipt of that report

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## NAVIGABLE CAMERA ARRAY AND VIEWER THEREFORE

### CROSS REFERENCE TO RELATED APPLICATIONS

The present application claims the benefit of United States Provisional Application Serial No. 60/285,201, filed April 20, 2001, entitled NAVIGABLE CAMERA ARRAY AND VIEWER THEREFORE, hereby incorporated herein by reference.

### BACKGROUND OF THE INVENTION

#### 1. Field Of The Invention

The present invention relates to a telepresence system and, more particularly, to a navigable camera array telepresence system and method of using same for comparing two or more images.

#### 2. Description Of Related Art

In general, a need exists for the development of telepresence systems suitable for use with static venues, such as museums, and dynamic venues or events, such as a music concerts. The viewing of such venues is limited by time, geographical location, and the viewer capacity of the venue. For example, potential visitors to a museum may be prevented from viewing an exhibit due to the limited hours the museum is open. Similarly, music concert producers must turn back fans due to the limited seating of an arena. In short, limited access to venues reduces the revenue generated.

In an attempt to increase the revenue stream from both static and dynamic venues, such venues have been recorded for broadcast or distribution. In some instances, dynamic venues are also broadcast live. While such broadcasting increases access to the venues, it involves considerable production effort. Typically, recorded broadcasts must be cut and

edited, as views from multiple cameras are pieced together. These editorial and production efforts are costly.

In some instances, the broadcast resulting from these editorial and production efforts provides viewers with limited enjoyment. Specifically, the broadcast is typically based on filming the venue from a finite number of predetermined cameras. Thus, the broadcast contains limited viewing angles and perspectives of the venue. Moreover, the viewing angles and perspectives presented in the broadcast are those selected by a producer or director during the editorial and production process; there is no viewer autonomy. Furthermore, although the broadcast is often recorded for multiple viewings, the broadcast has limited content life because each viewing is identical to the first. Because each showing looks and sounds the same, viewers rarely come back for multiple viewings.

A viewer fortunate enough to attend a venue in person will encounter many of the same problems. For example, a museum-goer must remain behind the barricades, viewing exhibits from limited angles and perspectives. Similarly, concert-goers are often restricted to a particular seat or section in an arena. Even if a viewer were allowed free access to the entire arena to videotape the venue, such a recording would also have limited content life because each viewing would be the same as the first. Therefore, a need exists for a telepresence system that preferably provides user autonomy while resulting in recordings with enhanced content life at a reduced production cost.

Apparently, attempts have been made to develop telepresence systems to satisfy some of the foregoing needs. One telepresence system is described in U.S. Patent No. 5,708,469 for Multiple View Telepresence Camera Systems Using A Wire Cage Which Surrounds A Polarity Of Multiple Cameras And Identifies The Fields Of View, issued January 13, 1998.

The system described therein includes a plurality of cameras, wherein each camera has a field of view that is space-contiguous with and at a right angle to at least one other camera. In other words, it is preferable that the camera fields of view do not overlap each other. A user interface allows the user to jump between views. In order for the user's view to move through the venue or environment, a moving vehicle carries the cameras.

This system, however, has several drawbacks. For example, in order for a viewer's perspective to move through the venue, the moving vehicle must be actuated and controlled. In this regard, operation of the system is complicated. Furthermore, because the camera views are contiguous, typically at right angles, changing camera views results in a discontinuous image.

Other attempts at providing a telepresence system have taken the form of a 360 degree camera systems. One such system is described in U.S. Patent No. 5,745,305 for Panoramic Viewing Apparatus, issued April 28 1998. The system described therein provides a 360 degree view of environment by arranging multiple cameras around a pyramid shaped reflective element. Each camera, all of which share a common virtual optical center, receives an image from a different side of the reflective pyramid. Other types of 360 degree camera systems employ a parabolic lens or a rotating camera.

Such 360 degree camera systems also suffer from drawbacks. In particular, such systems limit the user's view to 360 degrees from a given point perspective. In other words, 360 degree camera systems provide the user with a panoramic view from a single location. Only if the camera system was mounted on a moving vehicle could the user experience simulated movement through an environment.

U.S. Patent No. 5,187,571 for Television System For Displaying Multiple Views of A Remote Location issued February 16, 1993, describes a camera system similar to the 360 degree camera systems described above. The system described provides a user to select an arbitrary and continuously variable section of an aggregate field of view. Multiple cameras are aligned so that each camera's field of view merges contiguously with those of adjacent cameras thereby creating the aggregate field of view. The aggregate field of view may expand to cover 360 degrees. In order to create the aggregate field of view, the cameras' views must be contiguous. In order for the camera views to be contiguous, the cameras have to share a common point perspective, or vertex. Thus, like the previously described 360 degree camera systems, the system of U.S. Patent No. 5,187,571 limits a user's view to a single point perspective, rather than allowing a user to experience movement in perspective through an environment.

Also, with regard to the system of U.S. Patent No. 5,187,571, in order to achieve the contiguity between camera views, a relatively complex arrangement of mirrors is required. Additionally, each camera seemingly must also be placed in the same vertical plane.

Thus, a need still exists for an improved telepresence system that provides the ability to better simulate a viewer's actual presence in a venue, preferably in real time.

### **3. Summary of the Invention**

These and other needs are satisfied by the present invention. A viewer (e.g., electronic graphical user interface) according to one embodiment of the present invention is capable for use in selecting images and/or views from an array of cameras, each of which has an associated view of an environment and an associated output representing the view.

To summarize such a matrix viewer of one embodiment, it allows for the controlling of time and viewpoint, and warping of imagery in response to user inputs. The input to the matrix viewer can be either a raw set of pre-synthesized image data, or a set of original image data together with a set of flow-fields. In both cases the matrix viewer allows the user to navigate the data both in space and in time, with the use of 2 slider controls, a single graphical control (e.g., the four-quadrant button described above) and the like. In the present embodiment, the matrix viewer thus works generally according to the following four steps. Before being read into the matrix viewer, the data is first organized. The second step generally involves mapping the desired view given by the User Interface, and the next likely desired-view, onto the source data. The third step uses the mapping performed in the second step to fetch the data for the desired view and the next (anticipated) desired view into local memory. The forth step involves processing the data for the desired view as required, displaying the view on the screen, and continuing to fetch data for the next desired view.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an overall schematic of one embodiment of the present invention.

Figure 2a is a perspective view of a camera and a camera rail section of the array according to one embodiment of the present invention.

Figures 2b-2d are side plan views of a camera and a camera rail according to one embodiment of the present invention.

Figure 2e is a top plan view of a camera rail according to one embodiment of the present invention.

Figure 3 is a perspective view of a portion of the camera array according to one embodiment of the present invention.

Figure 4 is a perspective view of a portion of the camera array according to an alternate embodiment of the present invention.

Figure 5 is a flowchart illustrating the general operation of the user interface according to one embodiment of the present invention.

Figure 6 is a flowchart illustrating in detail a portion of the operation shown in Figure 5.

Figure 7a is a perspective view of a portion of one embodiment of the present invention illustrating the arrangement of the camera array relative to objects being viewed.

Figures 7b-7g illustrate views from the perspectives of selected cameras of the array in Figure 7a.

Figure 8 is a schematic view of an alternate embodiment of the present invention.

Figure 9 is a schematic view of a server according to one embodiment of the present invention.

Figure 10 is a schematic view of a server according to an alternate embodiment of the present invention.

Figure 11 is a top plan view of an alternate embodiment of the present invention.

Figure 12 is a flowchart illustrating in detail the image capture portion of the operation of the embodiment shown in Figure 11.

Figure 13 is a schematic illustrating an array of one embodiment of the present invention.

Figure 14 is flowchart illustrating the image capture process of one embodiment of the present invention.

Figure 15 is a schematic illustrating the logical arrangement of frames of an image according to one embodiment of the present invention.

Figure 16 is a flowchart illustrating the playback process of one embodiment of the present invention.

Figure 17 is a schematic view representing a display according to one embodiment of the present invention.

Figure 18a-c are schematics illustrating the logical relationship among frames according to one embodiment of the present invention.

Figure 19 is a schematic illustrating the logical arrangement of frames according to one embodiment of the present invention.

Figure 20 is a flowchart illustrating the process of harmonizing the duration of images according to one embodiment of the present invention.

Figure 21 is a schematic of a viewer according to one embodiment of the present invention.

Figure 22 is a schematic illustrating system components according to one embodiment of the present invention.

Figure 23 is a schematic illustrating the processing flow associated with capturing images, creating “on-the-fly” tweened images and making such images available to a viewer according to one embodiment of the present invention.

Figure 24 is a schematic illustrating the processing flow associated with capturing images, creating “pre-tweened” images and making such images available to a viewer according to one embodiment of the present invention.

## DESCRIPTION OF EMBODIMENTS

### 1. General Overview

The present invention relates to a viewer for use in selecting images and/or cameras of a telepresence system, such as that disclosed in International Application Serial No. PCT/US00/28652, assigned to Kewazinga Corporation (the "Kewazinga Application"), hereby incorporated herein by reference. As illustrated in the Kewazinga Application, in preferred embodiments, the telepresence system includes an array of cameras, the outputs of which are electronically provided to one or more users, each with its own viewer, in response to user inputs, such that the users can simultaneously and independently navigate through the array.

In certain preferred embodiment, the outputs of these microcameras are linked by tiny (less than half the width of a human hair) Vertical Cavity Surface Emitting Lasers (VCSELS) to optical fibers, fed through area net hubs, buffered on server arrays or server farms (either for recording or (instantaneous) relay) and sent to viewers at remote terminals, interactive wall screens, or mobile image appliances (like Virtual Retinal Displays). Each remote viewer, through an intuitive graphical user interface (GUI), can navigate effortlessly through the environment, enabling seamless movement through the event.

This involves a multiplexed, electronic switching process (invisible to the viewer) which moves the viewer's point perspective from camera to camera. Rather than relying, per se, on physically moving a camera through space, the system uses the multiplicity of positioned cameras to move the viewer's perspective from camera node to adjacent camera node in a way that provides the viewer with a sequential visual and acoustical path throughout the extent of the array. This allows the viewer to fluidly track or dolly through a

3-dimensional remote environment, to move through an event and make autonomous real-time decisions about where to move and when to linger.

Instead of investing the viewer with the capacity to physically move a robotic camera, which would immediately limit the number of viewers that could simultaneously control their own course and navigate via storage nodes containing images of an environment associated with a pre-existing array of cameras. The user can move around the environment in any direction -- clockwise or counterclockwise, up, down, closer to or further away from the environment, or some combination thereof. Moreover, image output mixing, such as mosaicing and tweening, effectuates seamless motion throughout the environment.

## **2. Detailed Description Of Embodiments**

Certain embodiments of the present invention will now be described in greater detail with reference to the drawings. It is understood that the operation and functionality of many of the components of the embodiments described herein are known to one skilled in the art and, as such, the present description does not go into detail into such operative and functionality.

A telepresence system 100 according to the present invention is shown in Fig. 1. The telepresence system 100 generally includes an array 10 of cameras 14 coupled to a server 18, which in turn is coupled to one or more users 22 each having a user interfaced/display device 24. As will be understood to one skilled in the art, the operation and functionality of the embodiment described herein is provided, in part, by the server and user interface/display device. While the operation of these components is not described by way of particular code listings or logic diagrams, it is to be understood that one skilled in the art will be able to arrive at suitable implementations based on the functional and operational details provided

herein. Furthermore, the scope of the present invention is not to be construed as limited to any particular code or logic implementation.

In the present embodiment, the camera array 10 is conceptualized as being in an X, Z coordinate system. This allows each camera to have an associated, unique node address comprising an X, and Z coordinate (X, Z). In the present embodiment, for example, a coordinate value corresponding to an axis of a particular camera represents the number of camera positions along that axis the particular camera is displaced from a reference camera. In the present embodiment, from the user's perspective the X axis runs left and right, and the Z axis runs down and up. Each camera 14 is identified by its X, Z coordinate. It is to be understood, however, that other methods of identifying cameras 14 can be used. For example, other coordinate systems, such as those noting angular displacement from a fixed reference point as well as coordinate systems that indicate relative displacement from the current camera node may be used. In another alternate embodiment, the array is three dimensional, located in an X, Y, Z coordinate system.

The array 10 comprises a plurality of rails 12, each rail 12 including a series of one or more cameras 14. The output from the cameras 14 are coupled to the server 18 by means of local area hubs 16. The local area hubs 16 gather the outputs and, when necessary, amplify the outputs for transmission to the server 18. In an alternate embodiment, the local area hubs 16 multiplex the outputs for transmission to the server 18. Although the figure depicts the communication links 15 between the cameras 14 and the server 18 as being hardwired, it is to be understood that wireless links may be employed. Thus, it is within the scope of the present invention for the communication links 15 to take the form of fiber optics, cable, satellite, microwave transmission, internet, and the like.

Also coupled to the server 18 is an electronic storage device 20. The server 18 transfers the outputs to the electronic storage device 20. The electronic (mass) storage device 20, in turn, transfers each camera's output onto a storage medium or means, such as CD-ROM, DVD, fluorescent multilayered disk (FMD), tape, platter, disk array, or the like. The output of each camera 14 is stored in particular locations on the storage medium associated with that camera 14 or is stored with an indication to which camera 14 each stored output corresponds. For example, the output of each camera 14 is stored in contiguous locations on a separate disk, tape, CD-ROM, or platter. As is known in the art, the camera output may be stored in a compressed format, such as JPEG, which is a standard format for storing still color and grayscale photographs in bitmap form, MPEG1, which is a standard format for storing video output with a resolution of 30 frames per second, MPEG2, which is a standard format for storing video output with a resolution of 60 frames per second (typically used for high bandwidth applications such as HDTV and DVD-ROMs), and the like. Having stored each output allows a user to later view the environment over and over again, each time moving through the array 10 in a new path, as described below. In some embodiments of the present invention, such as those providing only real-time viewing, no storage device is required.

As will be described in detail below, the server 18 receives output from the cameras 14 in the array. The server 18 processes these outputs for either storage in the electronic storage device 20, transmission to the users 22 or both.

It is to be understood that although the server 18 is configured to provide the functionality of the system 100 in the present embodiment, it is to be understood that other processing elements may provide the functionality of the system 100. For example, in

alternate embodiments, the user interface device is a personal computer programmed to interpret the user input and transmit an indication of the desired current node address, buffer outputs from the array, and provide other of the described functions.

As shown, the system 100 can accommodate (but does not require) multiple users 22. Each user 22 has associated therewith a user interface device including a user display device (collectively 24). For example, user 22-1 has an associated user interface device and a user display device in the form of a computer 24-1 having a monitor and a keyboard. User 22-2 has associated therewith an interactive wall screen 24-2 which serves as a user interface device and a user display device. The user interface device and the user display device of user 22-3 includes a mobile audio and image appliance 24-3. A digital interactive TV 24-4 is the user interface device and user display device of user 22-4. Similarly, user 22-5 has a voice recognition unit and monitor 24-5 as the user interface and display devices. It is to be understood that the foregoing user interface devices and user display devices are merely exemplary; for example, other interface devices include a mouse, touch screen, biofeedback devices, as well as those identified in U.S. Provisional Patent Application Serial No. 60/080,413 and the like.

As described in detail below, each user interface device 24 has associated therewith user inputs. These user inputs allow each user 22 to move or navigate independently through the array 10. In other words, each user 22 enters inputs to generally select which camera outputs are transferred to the user display device. Preferably, each user display device includes a graphical representation of the array 10. The graphical representation includes an indication of which camera in the array the output of which is being viewed. The user inputs allow each user to not only select particular cameras, but also to select relative movement or

navigational paths through the array 10. It is to be understood that as used herein a path is defined by both cameras and time. As such, two users navigating through the same series of cameras may navigate different paths, provided the users do not access all cameras simultaneously. In other words, a linear series of plurality of cameras provides for a plurality of paths.

As shown in Fig. 1, each user 22 may be coupled to the server 18 by an independent communication link. Furthermore, each communication link may employ different technology. For example, in alternate embodiments, the communication links include an internet link, a microwave signal link, a satellite link, a cable link, a fiber optic link, a wireless link, and the like.

It is to be understood that the array 10 provides several advantages. For example, because the array 10 employs a series of cameras 14, no individual camera, or the entire array 10 for that matter, need be moved in order to obtain a seamless view of the environment. Instead, the user navigates through the array 10, which is strategically placed through and around the physical environment to be viewed. Furthermore, because the cameras 14 of the array 10 are physically located at different points in the environment to be viewed, a user is able to view changes in perspective, a feature unavailable to a single camera that merely changes focal length.

(a)      Cameras

It is to be understood that the present invention does not depend upon any particular type of camera and as such, includes in alternate embodiments, analog or digital, video or still, or full size or microcameras--microlenses mounted on thumbnail-sized CMOS active pixel sensor (APS) microchips. The video chips used in microcameras may be CMOS, CCD

and the like, and are produced in a mainstream manufacturing process, by several companies, including Photobit, Pasadena, CA; Sarnoff Corporation, Princeton, NJ; and VLSI Vision, Ltd., Edinburgh, Scotland.

One specific suitable cameras is the analog color CCD camera manufactured by Sanyo Electric Co. Ltd. under the tradename VCC-5974. As will be appreciated by those skilled in the art, use of such an analog camera is in conjunction with video capture boards, such as those provided by the Matrox Electronic Systems under the tradename Meteor-II, which includes an analog to digital converter for converting analog NTSC video. In various embodiments involving video, the capture boards also receive a video synchronizing signal, noted below, so that the output of each camera is synchronize, with each captured frame of one camera corresponding to that of the other. From the capture boards, the camera output is then provided to one or more servers or processing elements for processing.

#### (b) Structure of the Array

The structure of the array 10 will now be described in greater detail with reference to Figs. 2a-2e. In general, the camera array 10 of the present embodiment comprises a series of modular rails 12 carrying cameras 14. The structure of the rails 12 and cameras 14 will now be discussed in greater detail with reference to Figs. 2a through 2d. Each camera 14 includes registration pins 34. In one embodiment, the cameras 14 utilize VCSELs to transfer their outputs to the rail 12. It is to be understood that the present invention is not limited to any particular type of camera 14, however, or even to an array 10 consisting of only one type of camera 14.

Each rail 12 includes two sides, 12a, 12b, at least one of which 12b is hingeably connected to the base 12c of the rail 12. The base 12c includes docking ports 36 for

receiving the registration pins 34 of the camera 14. When the camera 14 is seated on a rail 12 such that the registration pins 34 are fully engaged in the docking ports 36, the hinged side 12b of the rail 12 is moved against the base 32 of the camera 14, thereby securing the camera 14 to the rail 12.

Each rail 12 further includes a first end 38 and a second end 44. The first end 38 includes, in the present embodiment, two locking pins 40 and a protected transmission relay port 42 for transmitting the camera outputs. The second end 44 includes two guide holes 46 for receiving the locking pins 40, and a transmission receiving port 48. Thus, the first end 38 of one rail 12 is engageable with a second end 44 of another rail 12. Therefore, each rail 12 is modular and can be functionally connected to another rail to create the array 10.

Once the camera 14 is securely seated to the rail 12, the camera 14 is positioned such that the camera output may be transmitted via a cable or VCSEL to the rail 12. Each rail 12 includes communication paths for transmitting the output from each camera 14.

Alternatively, a cable couples each camera to the server.

Although the array 10 is shown having a particular configuration, it is to be understood that virtually any configuration of rails 12 and cameras 14 is within the scope of the present invention. For example, the array 10 may be a linear array of cameras 14, a 2-dimensional array of cameras 14, a 3-dimensional array of cameras 14, or any combination thereof. Furthermore, the array 10 need not be comprised solely of linear segments, but rather may include curvilinear sections.

Furthermore, in an alternate embodiment individual rails support a single camera and include varying degree of freedom extension spacers on either end of the rail to change the spacing between cameras or change the angle between adjacent cameras. These spacers

comprise linear or rotary actuators or electrostrictive polymers controlled by one of the system servers.

The array 10 is supported by any of a number of support means. For example, the array 10 can be fixedly mounted to a wall or ceiling; the array 10 can be secured to a moveable frame that can be wheeled into position in the environment or supported from cables.

Fig. 3 illustrates an example of a portion of the array 10. As shown, the array 10 comprises five rows of rails 12a, through 12e. Each of these rails 12a-12e is directed towards a central plane, which substantially passes through the center row 12c. Consequently, for any object placed in the same plane as the middle row 12c, a user would be able to view the object essentially from the bottom, front, and top.

As noted above, the rails 12 of the array 10 need not have the same geometry. For example, some of the rails 12 may be straight while others may be curved. For example, Fig. 4 illustrates the camera alignment that results from utilizing curved rails. It should be noted that rails in Fig. 4 have been made transparent so that the arrangement of cameras 14 may be easily seen.

In an alternate embodiment, each rail is configured in a step-like fashion or an arc with each camera above (or below) and in front of a previous camera. In such an arrangement, the user has the option of moving forward through the environment.

It is to be understood that the spacing of the cameras 14 depends on the particular application, including the objects being viewed, the focal length of the cameras 14, and the speed of movement through the array 10. In general, the closer the cameras and the greater the overlap in views, the more seamless the transition between camera views. In one

embodiment the distance between cameras 14 can be approximated by analogy to the distance between exposed frames taken by a motion picture camera dollying linearly through an environment. In general, the speed of movement of the projector through the environment divided by the frames exposed per unit of time results in a frame-distance ratio.

For example, as shown by the following equations, in some applications a frame is taken ever inch. A conventional movie camera records twenty-four frames per second. When such a camera is moved linearly through an environment at two feet per second, a frame is taken approximately every inch.

$$\frac{2 \text{ ft}}{\text{sec}} \div \frac{24 \text{ frames}}{\text{sec}} = \frac{2 \text{ ft}}{24 \text{ frames}} = \frac{1 \text{ ft}}{12 \text{ frames}} = \frac{12 \text{ inches}}{12 \text{ frames}}$$

1 inch = 1 frame per inch.

1 frame

A frame of the projector is analogous to a camera 14 in the present invention. Thus, where one frame exposed per inch results in a movie having a seamless view of the environment, so too does one camera 14 per inch. Thus, in one embodiment of the present invention the cameras 14 are spaced approximately one inch apart, thereby resulting in a seamless view of the environment.

In alternate embodiments, the spacing between cameras is greater than one inch, provided the fields of view of adjacent cameras overlap. Again, the greater the degree of overlap, the more seamless the progression between adjacent camera views.

As described in greater detail below, the spacing between cameras may be further increased by generating synthetic or mixed images between contiguous cameras. Furthermore, the linear spacing between cameras becomes less important in a curved array,

where the angular displacement between cameras is more important. For example, in one embodiment, the array is in a 180 degree arc, with cameras placed at five degree intervals, directed towards the center of the arc. As the radius of the arc increases, the linear distance between the cameras also increase; however, the angular displacement, five degrees, and the overlap in fields of view remain the same. Because the overlap in field of view remains, the system maintains the seamless progression from camera to adjacent camera.

In one embodiment the array comprises an arc of cameras. The arc extends 110 degrees, with a radius of nine feet, and the cameras placed at approximately seven and a half degree intervals around the arc. In another embodiment the arc has a radius of fifteen feet, with the cameras located every sixteen inches.

In certain embodiments, it is useful to calibrate the cameras, aligning them in the same horizontal and vertical planes. Such calibration is accomplished in various embodiments using lasers directed from each camera, a grid superimposed on each camera view and the like to align each camera respective to a reference point.

### 3. Navigation Through the System

The general operation of the present embodiment will now be described with reference to Fig. 5 and continuing reference to Figure 1. As shown in step 110, the user is presented with a predetermined starting view of the environment corresponding to a starting camera. It is to be understood that the operation of the system is controlled, in part, by software residing in the server. As noted above, the system associates each camera in the array with a coordinate. Thus, the system is able to note the coordinates of the starting camera node. The camera output and, thus the corresponding view, changes only upon receiving a user input.

When the user determines that they want to move or navigate through the array, the user enters a user input through the user interface device 24. As described below, the user inputs of the present embodiment generally include moving to the right, to the left, up, or down in the array. Additionally, a user may jump to a particular camera in the array. In alternate embodiments, a subset of these or other inputs, such as forward, backward, diagonal, over, and under, are used. The user interface device, in turn, transmits the user input to the server in step 120.

Next, the server receives the user input in step 130 and proceeds to decode the input. In the present embodiment, decoding the input generally involves determining whether the user wishes to move to the right, to the left, up, or down in the array.

On the other hand, if the received user input does not correspond to backward, then the server 18 proceeds to determine whether the input corresponds to moving to the user's right in the array 10. This determination is shown in step 140. If the received user input does correspond to moving to the right, the current node address is incremented along the X axis in step 150 to obtain an updated address.

If the received user input does not correspond to moving to the right in the array, the server 18 then determines whether the input corresponds to moving to the user's left in the array 10 in step 160. Upon determining that the input does correspond to moving to the left, the server 18 then decrements the current node address along the X axis to arrive at the updated address. This is shown in step 170.

If the received user input does not correspond to either moving to the right or to the left, the server 18 then determines whether the input corresponds to moving up in the array. This determination is made in step 180. If the user input corresponds to moving up, in step

190, the server 18 increments the current node address along the Z axis, thereby obtaining an updated address.

Next, the server 18 determines whether the received user input corresponds to moving down in the array 10. This determination is made in step 200. If the input does correspond to moving down in the array 10, in step 210 the server 18 decrements the current node address along the Z axis.

Lastly, in step 220 the server 18 determines whether the received user input corresponds to jumping or changing the view to a particular camera 14. As indicated in Figure 5, if the input corresponds to jumping to a particular camera 14, the server 18 changes the current node address to reflect the desired camera position. Updating the node address is shown as step 230. In an alternate embodiment, the input corresponds to jumping to a particular position in the array 10, not identified by the user as being a particular camera but by some reference to the venue, such as stage right.

It is to be understood that the server 18 may decode the received user inputs in any of a number of ways, including in any order. For example, in an alternate embodiment the server 18 first determines whether the user input corresponds to up or down. In another alternate, preferred embodiment, user navigation includes moving forward, backward, to the left and right, and up and down through a three dimensional array.

If the received user input does not correspond to any of the recognized inputs, namely to the right, to the left, up, down, or jumping to a particular position in the array 10 then in step 240, the server 18 causes a message signal to be transmitted to the user display device 24, causing a message to be displayed to the user 22 that the received input was not

understood. Operation of the system 100 then continues with step 120, and the server 18 awaits receipt of the next user input.

After adjusting the current node address, either by incrementing or decrementing the node address along an axis or by jumping to a particular node address, the server 18 proceeds in step 250 to adjust the user's view. Once the view is adjusted, operation of the system 100 continues again with step 120 as the server 18 awaits receipt of the next user input.

In an alternate embodiment, the server 18 continues to update the node address and adjust the view based on the received user input. For example, if the user input corresponded to "moving to the right", then operation of the system 100 would continuously loop through steps 140, 150, and 250, checking for a different input. When the different input is received, the server 18 continuously updates the view accordingly.

It is to be understood that the foregoing user inputs, namely, to the right, to the left, up, and down, are merely general descriptions of movement through the array. Although the present invention is not so limited, in the present preferred embodiment, movement in each of these general directions is further defined based upon the user input.

Accordingly, Fig. 6 is a more detailed diagram of the operation of the system according to steps 140, 150, and 250 of Fig. 5. Moreover, it is to be understood that while Fig. 6 describes more detailed movement one direction i.e., to the right, the same detailed movement can be applied in any other direction. As illustrated, the determination of whether the user input corresponds to moving to the right actually involves several determinations. As described in detail below, these determinations include moving to the right through the array 10 at different speeds, moving to the right into a composited additional source output at different speeds, and having the user input overridden by the system 100.

The present invention allows a user 22 to navigate through the array 10 at the different speeds. Depending on the speed (i.e. number of camera nodes transversed per unit of time) indicated by the user's input, such as movement of a pointing device (or other interface device), the server 18 will apply an algorithm that controls the transition between camera outputs either at critical speed ( $n$  nodes/per unit of time), under critical speed ( $n-1$  nodes/per unit of time), or over critical speed ( $n + 1$  nodes/per unit of time).

It is to be understood that speed of movement through the array 10 can alternatively be expressed as the time to switch from one camera 14 to another camera 14.

Specifically, as shown in step 140a, the server 18 makes the determination whether the user input corresponds to moving to the right at a critical speed. The critical speed is preferably a predetermined speed of movement through the array 10 set by the system operator or designer depending on the anticipated environment being viewed. Further, the critical speed depends upon various other factors, such as focal length, distance between cameras, distance between the cameras and the viewed object, and the like. The speed of movement through the array 10 is controlled by the number of cameras 14 traversed in a given time period. Thus, the movement through the array 10 at critical speed corresponds to traversing some number, "n", camera nodes per millisecond, or taking some amount of time, "s", to switch from one camera 14 to another. It is to be understood that in the same embodiment the critical speed of moving through the array 10 in one dimension need not equal the critical speed of moving through the array in another dimension. Consequently, the server 18 increments the current node address along the X axis at  $n$  nodes per millisecond.

In the present preferred embodiment the user traverses twenty-four cameras 14 per second. As discussed above, a movie projector records twenty-four frames per second.

Analogizing between the movie projector and the present invention, at critical the user traverses (and the server **18** switches between) approximately twenty-four cameras **14** per second, or a camera **14** approximately every 0.04167 seconds.

As shown in Figure 6, the user **22** may advance not only at critical speed, but also at over the critical speed, as shown in step **140b**, or at under the critical speed, as shown in step **140c**. Where the user input “I” indicates movement through the array **10** at over the critical speed, the server **18** increments the current node address along the X axis by a unit of greater than n, for example, at  $n + 2$  nodes per millisecond. The step of incrementing the current node address at  $n + 1$  nodes per millisecond along the X axis is shown in step **150b**. Where the user input “I” indicates movement through the array **10** at under the critical speed, the server **18** proceeds to increment the current node address at a variable less than n, for example, at  $n - 1$  nodes per millisecond. This operation is shown as step **150c**.

#### 4. Scalable Arrays

The shape of the array **10** can also be electronically scaled and the system **100** designed with a “center of gravity” that will ease a user’s image path back to a “starting” or “critical position” node or ring of nodes, either when the user **22** releases control or when the system **100** is programmed to override the user’s autonomy; that is to say, the active perimeter or geometry of the array **10** can be pre-configured to change at specified times or intervals in order to corral or focus attention in a situation that requires dramatic shaping. The system operator can, by real-time manipulation or via a pre-configured electronic proxy sequentially activate or deactivate designated portions of the camera array **10**. This is of particular importance in maintaining authorship and dramatic pacing in theatrical or

entertainment venues, and also for implementing controls over how much freedom a user 22 will have to navigate through the array 10.

In the present embodiment, the system 100 can be programmed such that certain portions of the array 10 are unavailable to the user 22 at specified times or intervals. Thus, continuing with step 140d of Fig. 6, the server 18 makes the determination whether the user input corresponds to movement to the right through the array but is subject to a navigation control algorithm. The navigation control algorithm causes the server 18 to determine, based upon navigation control factors, whether the user's desired movement is permissible.

More specifically, the navigation control algorithm, which is programmed in the server 18, determines whether the desired movement would cause the current node address to fall outside the permissible range of node coordinates. In the present embodiment, the permissible range of node coordinates is predetermined and depends upon the time of day, as noted by the server 18. Thus, in the present embodiment, the navigation control factors include time. As will be appreciated by those skilled in the art, permissible camera nodes and control factors can be correlated in a table stored in memory.

In an alternate embodiment, the navigation control factors include time as measured from the beginning of a performance being viewed, also as noted by the server. In such an embodiment, the system operator can dictate from where in the array a user will view certain scenes. In another alternate embodiment, the navigation control factor is speed of movement through the array. For example, the faster a user 22 moves or navigates through the array, the wider the turns must be. In other alternate embodiments, the permissible range of node coordinates is not predetermined. In one embodiment, the navigation control factors and,

therefore, the permissible range, is dynamically controlled by the system operator who communicates with the server via an input device.

Having determined that the user input is subject to the navigation control algorithm, the server 18 further proceeds, in step 150d, to increment the current node address along a predetermined path. By incrementing the current node address along a predetermined path, the system operator is able to corral or focus the attention of the user 22 to the particular view of the permissible cameras 14, thereby maintaining authorship and dramatic pacing in theatrical and entertainment venues.

In an alternate embodiment where the user input is subject to a navigation control algorithm, the server 18 does not move the user along a predetermined path. Instead, the server 18 merely awaits a permissible user input and holds the view at the current node. Only when the server 18 receives a user input resulting in a permissible node coordinate will the server 18 adjust the user's view.

##### **5. Additional Source Output**

In addition to moving through the array 10, the user 22 may, at predetermined locations in the array 10, choose to leave the real world environment being viewed. More specifically, additional source outputs, such as computer graphic imagery, virtual world imagery, applets, film clips, and other artificial and real camera outputs, are made available to the user 22. In one embodiment, the additional source output is composited with the view of the real environment. In an alternate embodiment, the user's view transfers completely from the real environment to that offered by the additional source output.

More specifically, the additional source output is stored (preferably in digital form) in the electronic storage device 20. Upon the user 22 inputting a desire to view the additional

source output, the server 18 transmits the additional source output to the user interface/display device 24. The present embodiment, the server 18 simply transmits the additional source output to the user display device 24. In an alternate embodiment, the server 18 first composites the additional source output with the camera output and then transmits the composited signal to the user interface/display device 24.

As shown in step 140e, the server 18 makes the determination whether the user input corresponds to moving in the array into the source output. If the user 22 decides to move into the additional source output, the server 18 adjusts the view by substituting the additional source output for the updated camera output identified in either of steps 150a-d.

Once the current node address is updated in either of steps 150a-d, the server 18 proceeds to adjust the user's view in step 250. When adjusting the view, the server 18 "mixes" the existing or current camera output being displayed with the output of the camera 14 identified by the updated camera node address. Mixing the outputs is achieved differently in alternate embodiments of the invention. In the present embodiment, mixing the outputs involves electronically switching at a particular speed from the existing camera output to the output of the camera 14 having the new current node address.

It is to be understood that in this and other preferred embodiments disclosed herein, the camera outputs are synchronized. As is well known in the art, a synchronizing signal from a "sync generator" is supplied to the cameras and/or the processors capturing the camera output. The sync generator may take the form of those used in video editing and may comprise, in alternate embodiments, part of the server, the hub, and/or a separate component coupled to the array.

As described above, at critical speed, the server 18 switches camera outputs approximately at a rate of 24 per second, or one every 0.04167 seconds. If the user 22 is moving through the array 10 at under the critical speed, the outputs of the intermediate cameras 14 are each displayed for a relatively longer duration than if the user is moving at the critical speed. Similarly, each output is displayed for a relatively shorter duration when a user navigates at over the critical speed. In other words, the server 18 adjusts the switching speed based on the speed of the movement through the array 10.

Of course, it is to be understood that in a simplified embodiment of the present invention, the user may navigate at only the critical speed.

In another alternate embodiment, mixing the outputs is achieved by compositing the existing or current output and the updated camera node output. In yet another embodiment, mixing involves dissolving the existing view into the new view. In still another alternate embodiment, mixing the outputs includes adjusting the frame refresh rate of the user display device. Additionally, based on speed of movement through the array, the server may add motion blur to convey the realistic sense of speed.

In yet another alternate embodiment, the server causes a black screen to be viewed instantaneously between camera views. Such an embodiment is analogous to blank film between frames in a movie reel. Furthermore, although not always advantageous, such black screens reduce the physiologic "carrying over" of one view into a subsequent view.

It is to be understood that the user inputs corresponding to movements through the array at different speeds may include either different keystrokes on a keypad, different positions of a joystick, positioning a joystick in a given position for a predetermined length

of time, and the like. Similarly, the decision to move into an additional source output may be indicated by a particular keystroke, joystick movement, or the like.

In another embodiment, mixing may be accomplished by "mosaicing" the outputs of the intermediate cameras 14. U.S. Pat. No. 5,649,032 entitled System For Automatically Aligning Images To Form A Mosaic Image to Peter J. Burt et al. discloses a system and method for generating a mosaic from a plurality of images and is hereby incorporated by reference. The server 18 automatically aligns one camera output to another camera output, a camera output to another mosaic (generated from previously occurring camera output) such that the output can be added to the mosaic, or an existing mosaic to a camera output.

Once the mosaic alignment is complete, the present embodiment utilizes a mosaic composition process to construct (or update) a mosaic. The mosaic composition comprises a selection process and a combination process. The selection process automatically selects outputs for incorporation into the mosaic and may include masking and cropping functions to select the region of interest in a mosaic. Once the selection process selects which output(s) are to be included in the mosaic, the combination process combines the various outputs to form the mosaic. The combination process applies various output processing techniques, such as merging, fusing, filtering, output enhancement, and the like, to achieve a seamless combination of the outputs. The resulting mosaic is a smooth view that combines the constituent outputs such that temporal and spatial information redundancy are minimized in the mosaic. In one embodiment of the present invention, the mosaic may be formed as the user moves through the system and the output image displayed close to real time. In another embodiment, the system may form the mosaic from a predetermined number of outputs or

during a predetermined time interval, and then display the images pursuant to the user's navigation through the environment.

In yet another embodiment, the server 18 enables the output to be mixed by a "tweening" process. One example of the tweening process is disclosed in U.S. Pat. No. 5,259,040 entitled Method For Determining Sensor Motion And Scene Structure And Image Processing System Therefor to Keith J. Hanna, herein incorporated by reference. Tweening enables the server 18 to process the structure of a view from two or more camera outputs of the view.

Applying the Hanna patent to the telepresence method/system herein, tweening is now described. The server monitors the movement among the intermediate cameras 14 through a scene using local scene characteristics such as brightness derivatives of a pair of camera outputs. A global camera output movement constraint is combined with a local scene characteristic constancy constraint to relate local surface structures with the global camera output movement model and local scene characteristics. The method for determining a model for global camera output movement through a scene and scene structure model of the scene from two or more outputs of the scene at a given image resolution comprises the following steps:

- (a) setting initial estimates of local scene models and a global camera output movement model;
- (b) determining a new value of one of the models by minimizing the difference between the measured error in the outputs and the error predicted by the model;
- (c) resetting the initial estimates of the local scene models and the image sensor motion model using the new value of one of the models determined in step (b);

(d) determining a new value of the second of the models using the estimates of the models determined in step (b) by minimizing the difference between the measured error in the outputs and the error predicted by the model;

(e) warping one of the outputs towards the other output using the current estimates of the models at the given image resolution; and

(f) repeating steps (b), (c), (d) and (e) until the differences between the new values of the models and the values determined in the previous iteration are less than a certain value or until a fixed number of iterations have occurred.

It should be noted that where the Hanna patent effectuates the tweening process by detecting the motion of an image sensor (e.g., a video camera), an embodiment of the present invention monitors the user movement among live cameras or storage nodes.

As will be appreciated by those skilled in the art based on the present disclosure, other existing techniques may be applied to the mixing or tweening of outputs in any of the embodiments based on the teachings herein. Such other techniques are described in U.S. Patents U.S. 5,649,032, for System For Automatically Aligning Images To Form A Mosaic Image; U.S. 5,629,988, for System And Method For Electronic Image Stabilization; U.S. 5,581,629, for Method For Estimating The Location Of An Image Target Region From Tracked Multiple Image Landmark Regions; U.S. 5,488,674, for Method For Fusing Images And Apparatus Therefor; U.S. 5,067,014 Three-Frame Technique For Analyzing Two Motions In Successive Image Frames Dynamically, each of which are hereby incorporated by reference.

In an alternate embodiment, although not always necessary, to ensure a seamless progression of views, the server 18 also transmits to the user display device 24 outputs from

some or all of the intermediate cameras, namely those located between the current camera node and the updated camera node. Such an embodiment will now be described with reference to Figs. 7a-7g. Specifically, Fig. 7a illustrates a curvilinear portion of an array 10 that extends along the X axis or to the left and right from the user's perspective. Thus, the coordinates that the server 18 associates with the cameras 14 differ only in the X coordinate. More specifically, for purposes of the present example, the cameras 14 can be considered sequentially numbered, starting with the left-most camera 14 being the first, i.e., number "1". The X coordinate of each camera 14 is equal to the camera's position in the array. For illustrative purposes, particular cameras will be designate 14-X, where X equals the camera's position in the array 10 and, thus, its associated X coordinate.

In general, Figs. 7a-7g illustrate possible user movement through the array 10. The environment to be viewed includes three objects 602, 604, 606, the first and second of which include numbered surfaces. As will be apparent, these numbered surface allow a better appreciation of the change in user perspective.

In Fig. 7a, six cameras 14-2, 14-7, 14-11, 14-14, 14-20, 14-23 of the array 10 are specifically identified. The boundaries of each camera's view is identified by the pair of lines 14-2a, 14-7a, 14-11a, 14-14a, 14-20a, 14-23a, radiating from each identified camera 14-2, 14-7, 14-11, 14-14, 14-20, 14-23, respectively. As described below, in the present example the user 22 navigates through the array 10 along the X axis such that the images or views of the environment are those corresponding to the identified cameras 14-2, 14-7, 14-11, 14-14, 14-20, 14-23.

The present example provides the user 22 with the starting view from camera 14-2. This view is illustrated in Fig. 7b. The user 22, desiring to have a better view of the object

702, pushes the "7" key on the keyboard. This user input is transmitted to and interpreted by the server 18.

Because the server 18 has been programmed to recognize the "7" key as corresponding to moving or jumping through the array to camera 14-7. The server 18 changes the X coordinate of the current camera node address to 7, selects the output of camera 14-7, and adjusts the view or image sent to the user 22. Adjusting the view, as discussed above, involves mixing the outputs of the current and updated camera nodes. Mixing the outputs, in turn, involves switching intermediate camera outputs into the view to achieve the seamless progression of the discrete views of cameras 14-2 through 14-7, which gives the user 22 the look and feel of moving around the viewed object. The user 22 now has another view of the first object 702. The view from camera 14-7 is shown in Fig. 7c. As noted above, if the jump in camera nodes is greater than a predetermined limit, the server 18 would omit some or all of the intermediate outputs.

Pressing the "right arrow" key on the keyboard, the user 22 indicates to the system 100 a desire to navigate to the right at critical speed. The server 18 receives and interprets this user input as indicating such and increments the current camera node address by n=4. Consequently, the updated camera node address is 14-11. The server 18 causes the mixing of the output of camera 14-11 with that of camera 14-7. Again, this includes switching into the view the outputs of the intermediate cameras (i.e., 14-8, 14-9, and 14-10) to give the user 22 the look and feel of navigating around the viewed object. The user 22 is thus presented with the view from camera 14-11, as shown in Fig. 7d.

Still interested in the first object 702, the user 22 enters a user input, for example, "alt-right arrow," indicating a desire to move to the right at less than critical speed.

Accordingly, the server 18 increments the updated camera node address by n-1 nodes, namely 3 in the present example, to camera 14-14. The outputs from cameras 14-11 and 14-14 are mixed, and the user 22 is presented with a seamless view associated with cameras 14-11 through 14-14. Fig. 7e illustrates the resulting view of camera 14-14.

With little to see immediately after the first object 702, the user 22 enters a user input such as "shift-right arrow," indicating a desire to move quickly through the array 10, i.e., at over the critical speed. The server 18 interprets the user input and increments the current node address by n+2, or 6 in the present example. The updated node address thus corresponds to camera 14-20. The server 18 mixes the outputs of cameras 14-14 and 14-20, which includes switching into the view the outputs of the intermediate cameras 14-15 through 14-19. The resulting view of camera 14-20 is displayed to the user 22. As shown in Fig. 7f, the user 22 now views the second object 704.

Becoming interested in the third object 704, the user 22 desires to move slowly through the array 10. Accordingly, the user 22 enters "alt-right arrow" to indicate moving to the right at below critical speed. Once the server 18 interprets the received user input, it updates the current camera node address along the X axis by 3 to camera 14-23. The server 18 then mixes the outputs of camera 14-20 and 14-23, thereby providing the user 22 with a seamless progression of views through camera 14-23. The resulting view 14-23a is illustrated in Fig. 7g.

## 6. Other Data Devices

It is to be understood that devices other than cameras may be interspersed in the array. These other devices, such as motion sensors and microphones, provide data to the server(s) for processing. For example, in alternate embodiments output from motion sensors

or microphones are fed to the server(s) and used to scale the array. More specifically, permissible camera nodes (as defined in a table stored in memory) are those near the sensor or microphone having a desired output e.g., where there is motion or sound. As such, navigation control factors include output from other such devices. Alternatively, the output from the sensors or microphones are provided to the user.

An alternate embodiment in which the array of cameras includes multiple microphones interspersed among the viewed environment and the cameras will now be described with reference to Fig. 8. The system 800 generally includes an array of cameras 802 coupled to a server 804, which, in turn, is coupled to one or more user interface and display devices 806 and an electronic storage device 808. A hub 810 collects and transfers the outputs from the array 802 to the server 804. More specifically, the array 802 comprises modular rails 812 that are interconnected. Each rail 812 carries multiple cameras 814 and a microphone 816 centrally located at rail 812. Additionally, the system 800 includes microphones 818 that are physically separate from the array 802. The outputs of both the cameras 814 and microphones 816, 818 are coupled to the server 804 for processing.

In general, operation of the system 800 proceeds as described with respect to system 100 of Figures 1-2d and 5-6. Beyond the operation of the previously described system 100, however, the server 804 receives the sound output from the microphones 816, 818 and, as with the camera output, selectively transmits sound output to the user. As the server 804 updates the current camera node address and changes the user's view, it also changes the sound output transmitted to the user. In the present embodiment, the server 804 has stored in memory an associated range of camera nodes with a given microphone, namely the cameras 814 on each rail 810 are associated with the microphone 816 on that particular rail 810. In

the event a user attempts to navigate beyond the end of the array 802, the server 804 determines the camera navigation is impermissible and instead updates the microphone node output to that of the microphone 818 adjacent to the array 802.

In an alternate embodiment, the server 804 might include a database in which camera nodes in a particular area are associated with a given microphones. For example, a rectangular volume defined by the (X, Y, Z) coordinates (0,0,0), (10,0,0), (10,5,0), (0,5,0), (0,0,5), (10,0,5), (10,5,5) and (0,5,5) are associated with a given microphone. It is to be understood that selecting one of the series of microphones based on the user's position (or view) in the array provides the user with a sound perspective of the environment that coincides with the visual perspective.

It is to be understood that the server of the embodiments discussed above may take any of a number of known configurations. Two examples of server configurations suitable for use with the present invention will be described with reference to Figures 9 and 10.

Turning first to Figure 9, the server 902, electronic storage device 20, array 10, users (1,2,3, . . . N) 22-1 – 22-N, and associated user interface/display devices 24-1 – 24-N are shown therein.

The server 902 includes, among other components, a processing means in the form of one or more central processing units (CPU) 904 coupled to associated read only memory (ROM) 906 and a random access memory (RAM) 908. In general, ROM 906 is for storing the program that dictates the operation of the server 902, and the RAM 908 is for storing variables and values used by the CPU 904 during operation. Also coupled to the CPU 904 are the user interface/display devices 24. It is to be understood that the CPU may, in

alternate embodiments, comprise several processing units, each performing a discrete function.

Coupled to both the CPU 904 and the electronic storage device 20 is a memory controller 910. The memory controller 910, under direction of the CPU 904, controls accesses (reads and writes) to the storage device 20. Although the memory controller 910 is shown as part of the server 902, it is to be understood that it may reside in the storage device 20.

During operation, the CPU 904 receives camera outputs from the array 10 via bus 912. As described above, the CPU 904 mixes the camera outputs for display on the user interface/display device 24. Which outputs are mixed depends on the view selected by each user 22. Specifically, each user interface/display devices 24 transmits across bus 914 the user inputs that define the view to be displayed. Once the CPU 904 mixes the appropriate outputs, it transmits the resulting output to the user interface/display device 24 via bus 916. As shown, in the present embodiment, each user 22 is independently coupled to the server 902.

The bus 912 also carries the camera outputs to the storage device 20 for storage. When storing the camera outputs, the CPU 904 directs the memory controller 910 to store the output of each camera 14 in particular locations of memory in the storage device 20.

When the image to be displayed has previously been stored in the storage device 20, the CPU 904 causes the memory controller 910 to access the storage device 20 to retrieve the appropriate camera output. The output is thus transmitted to the CPU 904 via bus 918 where it is mixed. Bus 918 also carries additional source output to the CPU 904 for transmission to

the users 22. As with outputs received directly from the array 10, the CPU 904 mixes these outputs and transmits the appropriate view to the user interface/display device 24.

Figure 10 shows a server configuration according to an alternate embodiment of the present invention. As shown therein, the server 1002 generally comprises a control central processing unit (CPU) 1004, a mixing CPU 1006 associated with each user 22, and a memory controller 1008. The control CPU 1004 has associated ROM 1010 and RAM 1012. Similarly, each mixing CPU 1006 has associated ROM 1014 and RAM 1016.

To achieve the functionality described above, the camera outputs from the array 10 are coupled to each of the mixing CPUs 1 through N 1006-1, 1006-N via bus 1018. During operation, each user 22 enters inputs in the interface/display device 24 for transmission (via bus 1020) to the control CPU 1004. The control CPU 1004 interprets the inputs and, via buses 1022-1, 1022-N, transmits control signals to the mixing CPUs 1006-1, 1006-N instructing them which camera outputs received on bus 1018 to mix. As the name implies, the mixing CPUs 1006-1, 1006-N mix the outputs in order to generate the appropriate view and transmit the resulting view via buses 1024-1, 1024-N to the user interface/display devices 24-1, 24-N.

In an alternate related embodiment, each mixing CPU 1006 multiplexes outputs to more than one user 22. Indications of which outputs are to mixed and transmitted to each user 22 comes from the control CPU 1004.

The bus 1018 couples the camera outputs not only to the mixing CPUs 1006-1, 1006-N, but also to the storage device 20. Under control of the memory controller 1008, which in turn is controlled by the control CPU 1004, the storage device 20 stores the camera outputs in known storage locations. Where user inputs to the control CPU 1004 indicate a users' 22

desire to view stored images, the control CPU 1004 causes the memory controller 1008 to retrieve the appropriate images from the storage device 20. Such images are retrieved into the mixing CPUs 1006 via bus 1026. Additional source output is also retrieved to the mixing CPUs 1006-1, 1006-N via bus 1026. The control CPU 1004 also passes control signals to the mixing CPUs 1006-1, 1006-N to indicate which outputs are to be mixed and displayed.

In an embodiment analogous to that of Figure 10, the outputs of cameras are provided to networked (e.g., via an Ethernet) personal computers, for example one capture computer per pair of adjacent cameras and one control computer. In one embodiment, where analog video cameras are used, each capture computer also includes two video capture boards—one per camera coupled to the capture computer. Each capture computer also provides the mixing functionality, such as tweening, between the cameras coupled thereto. Furthermore, the control computer causes each capture computer to receive the output from a camera adjacent to one directly coupled to the capture computer so that capture computer may mix the outputs of the camera directly coupled to the capture computer and the adjacent camera. For example, if one capture computer is coupled to cameras “1” and “2”, and a second capture computer is coupled to cameras “3” and “4”, then the second capture computer would also receive the output of camera “2” so that such output could be mixed with that of adjacent camera “3”. Control computer coordinates the operation of the capture computers and other components as described herein.

## 7. Stereoscopic Views

It is to be understood that it is within the scope of the present invention to employ stereoscopic views of the environment. To achieve the stereoscopic view, the system retrieves from the array (or the electronic storage device) and simultaneously transmits to the

user at least portions of outputs from two cameras. The server processing element mixes these camera outputs to achieve a stereoscopic output. Each view provided to the user is based on such a stereoscopic output. In one stereoscopic embodiment, the outputs from two adjacent cameras in the array are used to produce one stereoscopic view. Using the notation of Figs. 7a-7g, one view is the stereoscopic view from cameras 14-1 and 14-2. The next view is based on the stereoscopic output of cameras 14-2 and 14-3 or two other cameras. Thus, in such an embodiment, the user is provided the added feature of a stereoscopic seamless view of the environment.

#### 8. Multiple Users

As described above, the present invention allows multiple users to simultaneously navigate through the array independently of each other. To accommodate multiple users, the systems described above distinguish between inputs from the multiple users and selects a separate camera output appropriate to each user's inputs. In one such embodiment, the server tracks the current camera node address associated with each user by storing each node address in a particular memory location associate with that user. Similarly, each user's input is differentiated and identified as being associated with the particular memory location with the use of message tags appended to the user inputs by the corresponding user interface device.

In an alternate embodiment, two or more users may choose to be linked, thereby moving in tandem and having the same view of the environment. In such an embodiment, each includes identifying another user by his/her code to serve as a "guide". In operation, the server provides the outputs and views selected by the guide user to both the guide and the

other user selecting the guide. Another user input causes the server to unlink the users, thereby allowing each user to control his/her own movement through the array.

## 9. Multiple Arrays

In certain applications, a user may also wish to navigate forward and backward through the environment, thereby moving closer to or further away from an object. Although it is within the scope of the present invention to use cameras with zoom capability, the use of a zoom lens would entail robotic control by a single user and preclude the simultaneous viewing of different fields of view positions at that camera node by multiple users. One embodiment that solves this problem of preventing multiple user from simultaneously viewing different fields of view from the same camera position in the array entails creating different field of view options at a single camera position. In alternate embodiments the different field of view options are created with clusters of cameras at each position in the array, each camera having a different field of view lens but substantially the same vertex in the array. In one embodiment, the cameras at the same position have essentially the same vertex by employing beam splitters and/or mirrors to enable the different field of view cameras to be physically positioned away from the vertex in the array, yet have each camera field of view from the same perspective or vertex. Where multiple cameras are used at a particular node in an array, each camera and its associated output has an address, a storage location where the camera outputs are being stored, and is accessible based on user inputs indicating which field of view or relative field of view (zoom in or zoom out) the user desires to receive. Additionally, it is to be understood that the use of such multiple cameras at a given node or location in the array may be used in any of the embodiments described herein.

Simply zooming towards an object, while simplifying the background and recompose of the scene, does not provide the visual cues, such as changing perspective lines, changing shadows and reflections, that actually moving forward through the environment provides. One such embodiment in which users can move dimensionally forward and backward through the environment with a changing image point perspective will now be described with respect to Fig. 11 and continuing reference to Fig. 1. As will be understood by those skilled in the art, the arrays described with reference to Fig. 11 may be used with any server, storage device and user terminals described herein.

Fig. 11 illustrates a top plan view of another embodiment enabling the user to move left, right, up, down, forward or backwards through the environment. A plurality of cylindrical arrays (**121-1 – 121-n**) of differing diameters comprising a series of cameras **14** may be situated around an environment comprising one or more objects **1200**, one cylindrical array at a time. Cameras **14** situated around the object(s) **1100** are positioned along an X and Z coordinate system. Accordingly, an array **12** may comprise a plurality of rings of the same circumference positioned at different positions (heights) throughout the z-axis to form a cylinder of cameras **14** around the object(s) **1100**. This also allows each camera in each array **12** to have an associated, unique storage node address comprising an X and Z coordinate – i.e., array1(X, Z). In the present embodiment, for example, a coordinate value corresponding to an axis of a particular camera represents the number of camera positions along that axis the particular camera is displaced from a reference camera. In the present embodiment, from the user's perspective, the X axis runs around the perimeter of an array **12**, and the Z axis runs down and up. Each storage node is associated with a camera view identified by its X, Z coordinate.

As described above, the outputs of the cameras 14 are coupled to one or more servers for gathering and transmitting the outputs to the server 18.

In one embodiment, because the environment is static, each camera requires only one storage location. The camera output may be stored in a logical arrangement, such as a matrix of n arrays, wherein each array has a plurality of (X,Z) coordinates. In one embodiment, the node addresses may comprise of a specific coordinate within an array – i.e.,  $\text{Array}_1(X_n, Z_n)$ ,  $\text{Array}_2(X_n, Z_n)$  through  $\text{Array}_n(X_n, Z_n)$ . As described below, users can navigate the stored images in much the same manner as the user may navigate through an environment using live camera images.

The general operation of one embodiment of inputting images in storage device 20 for transmission to a user will now be described with reference to Fig. 12 and continuing reference to Fig. 11. As shown in step 1210, a cylindrical array 12-1 is situated around the object(s) located in an environment 1100. The view of each camera 14 is transmitted to server 18 in step 1220. Next, in step 1220, the electronic storage device 20 of the server 18 stores the output of each camera 14 at the storage node address associated with that camera 14. Storage of the images may be effectuated serially, from one camera 14 at a time within the array 12, or by simultaneous transmission of the image data from all of the cameras 14 of each array 12. Once the output for each camera 14 of array 12-1 is stored, cylindrical array 12-1 is removed from the environment (step 1240). In step 1250, a determination is made as to the availability of additional cylindrical arrays 12 of differing diameters to those already situated. If additional cylindrical arrays 12 are desired, the process repeats beginning with step 1210. When no additional arrays 12 are available for situating around the environment,

the process of inputting images into storage devices 20 is complete (step 1260). At the end of the process, a matrix of addressable stored images exist.

Upon storing all of the outputs associated with the arrays 12-1 through 12-n, a user may navigate through the environment. Navigation is effectuated by accessing the input of the storage nodes by a user interface device 24. In the present embodiment, the user inputs generally include moving around the environment or object 1100 by moving to the left or right, moving higher or lower along the z-axis, moving through the environment closer or further from the object 1100, or some combination of moving around and through the environment. For example, a user may access the image stored in the node address Array<sub>3</sub>(0,0) to view an object from the camera previously located at coordinate (0,0) of Array<sub>3</sub>. The user may move directly forward, and therefore closer to the object 1100, by accessing the image stored in Array<sub>2</sub>(0,0) and then Array<sub>1</sub>(0,0). To move further away from the object and to the right and up, the user may move from the image stored in node address Array<sub>1</sub>(0,0) and access the images stored in node address Array<sub>2</sub>(1,1), followed by accessing the image stored in node address Array<sub>3</sub>(2,2), and so on. A user may, of course, move among arrays and/or coordinates by any increments changing the point perspective of the environment with each node. Additionally, a user may jump to a particular camera view of the environment. Thus, a user may move throughout the environment in a manner similar to that described above with respect to accessing output of live cameras. This embodiment, however, allows user to access images that are stored in storage nodes as opposed to accessing live cameras. Moreover, this embodiment provides a convenient system and method to allow a user to move forward and backward in an environment.

It should be noted that although each storage node is associated with a camera view identified by its X, Z coordinate of a particular array, other methods of identifying camera views and storage nodes can be used. For example, other coordinate systems, such as those noting angular displacement from a fixed reference point as well as coordinate systems that indicate relative displacement from the current camera node may be used. It should also be understood that the camera arrays 12 may be other shapes other than cylindrical. Moreover, it is not essential, although often advantageous, that the camera arrays 12 surround the entire environment.

It is to be understood that the foregoing user inputs, namely, move clockwise, move counter-clockwise, up, down, closer to the environment, and further from the environment, are merely general descriptions of movement through the environment. Although the present invention is not so limited, in the present preferred embodiment, movement in each of these general directions is further defined based upon the user input. Moreover the output generated by the server to the user may be mixed when moving among adjacent storage nodes associated with environment views (along the x axis, z axis, or among juxtaposed arrays) to generate seamless movement throughout the environment. Mixing may be accomplished by, but are not limited to, the processes described above.

As indicated above, an array according to the present invention may be used to capture virtually any image for any purpose. One particular use of one embodiment of the present invention is to compare multiple images. As will be appreciated from the following description, when used to compare images, the present invention can allow for a comparison from any one of multiple point perspectives at any given reference point of time. Exemplary embodiments, which will now be described with reference to Figs. 15-17, provide a training

aid that compares the images of the swings of two golfers -- a training professional and a player/trainee.

As shown in Fig. 13, the array is generally in the form of a geodesic dome 1305 having an opening for a golfer to enter and hit a ball. More specifically, the array extends approximately 270° in a horizontal band, 180° in a vertical band from side to side and 150° in a vertical band from the rear at the ground, forward towards the opening.

The array not only includes cameras 1310, but also lights 1315, a greenscreen background covering 1320, a greenscreen background flooring 1325, and a supporting rail structure 1330. As is known in the art, other color backgrounds can be used. The plurality of cameras 1310 populate the interior of the dome 1305 supported by the greenscreen 1320 and/or rails 1330. As described in greater detail below, the green covering 1320 and flooring 1325 allow for easier processing of the images.

As also described in detail below, the cameras 1310 can be logically organized in rows; for example, the lowest row 1335 can be designated  $\text{row}_0$ , the second row from the bottom 1340 can be designated  $\text{row}_1$ , the third row from the bottom 1345 can be designated  $\text{row}_2$ . Additionally, the cameras 1310 in each row can be logically numbered, for example, sequentially from the right of the array, clockwise to the left. As described below, such logical arrangement facilitates processing of images and navigation through the array. In alternate embodiments, the cameras 1310 are mounted in configurations other than rows, such as geometric or random patterns, preferably so that the image captured by one camera 1310 overlaps the image captured by each adjacent camera 1310.

Although only the array is depicted in Figure 13, it is to be understood that the array can be coupled to one or more processing elements, storage devices, user interface devices,

and other components according to any one of the configurations described above with reference to Figures 1 and 8-10 and equivalents thereto. In the present embodiment, the images of the professional's swing is stored in one storage device and the image of the trainee's swing is stored in a second storage device. In alternate embodiments, the images of the two swings are stored in different layers, levels or partitions within a single storage device, such as a fluorescent multi-layer disk. Each of the two storage devices are coupled in parallel to and can be accessed in parallel by the server. Furthermore, the cameras 1310 are coupled to the electronic storage devices so the images may be stored and the server is coupled to the storage devices so images can be retrieved from storage, processed and re-stored in the storage devices. A user interface device is also coupled to the server so the images can be transmitted to the user.

The capturing and storing of the images will now be described with reference to Figure 14. Once one of the golfers enters the dome 1305 and the system is activated, the system captures the image of the golfer's swing (step 1405). In the present embodiment each camera 1310 operates at approximately thirty frames per second. In an alternate embodiment, the cameras 1310 capture the image at sixty frames per second. The image from each camera 1310 and for each frame is then processed to separate the image from the background. More specifically, the server (or dedicated processor) mattes out the image from the solid background 1320 (step 1410). Such a process is generally known as bluescreening, matting, keying or chromakeying out the image and can be performed by any of a number of known processes, including those provided by the Ultimatte Corporation under the trade name ULTIMATTE, and by PixelCom J.V. under the trade name PRIMATTE. As will be appreciated by those skilled in the art, matting out the image is

preferable for better display of the images. The server then digitally stores the matted or keyed out image of each frame from each camera 1310 in an electronic storage device (step 1415).

Although not required, the present embodiment, the outputs (or images) captured in each frame of each camera 1310 are temporarily stored. The server then processes the temporarily stored frames to matte/key out the golfer's image from each frame and stores the matted/keyed out image, preferably writing over the original (non-keyed) frames. In an alternate embodiment, the server processes the frames, keying out the golfer's image, in real time. In such an embodiment, no temporary image need be stored. In another embodiment, no matting process is performed. Once the professional golfer's swing is captured, the system operation is repeated to capture and store the trainee's swing (step 1420).

Figure 15 depicts one example of a logical representation and addressing scheme of one golfer's swing as stored in one storage device without storing any mixed images. Taking thirty frames per second and the average golf swing lasting less than three seconds, approximately ninety frames will be stored for each camera. As logically shown, each frame from each camera is stored at a unique location or address in the storage device. In this embodiment, the first and second (right most) digits of the address indicate frame number, the third and fourth digits indicate camera number, and the fifth and sixth digits indicate row number. Thus, using the notation  $\text{row}_x(y)$  to denote the  $y^{\text{th}}$  camera of row  $x$  and the notation  $\text{frame}_z$  to denote the  $Z^{\text{th}}$  frame taken, the first frame --  $\text{frame}_1$  -- taken by the first camera in the first row --  $\text{row}_1(1)$  -- is stored at address 01 01 01. Similarly, the third frame --  $\text{frame}_3$  -- taken by the second camera in the second row --  $\text{row}_2(2)$  -- is stored at address 02 02 03.

It is to be understood that essentially any addressing scheme may be used for storing camera outputs, so long as the software playing back the images is capable of identifying the appropriate camera output in response to user inputs. In alternate embodiments the addresses can be represented in any notation, such as hexadecimal or binary, and the addresses may or may not be contiguous. Although not required, in the present embodiment, the same logical arrangement is used for the storage of the second golfer's swing in the second storage device. Furthermore, it should be understood that the foregoing discussion is a logical description of storing images for accessing certain portions thereof. Thus, in certain embodiments, the images are video streams, rather than separate, discrete frames.

Having described the capture and storage of the images, the playback of the images will now be described with reference to Figures 16 and 17 and continuing reference to Figures 13 and 15. As an initial step, the user selects playback on the user terminal (step 1605) and the playback begins. More specifically, the system begins by providing the user a default starting view of the professional and trainee (step 1610). In the present embodiment, the images of the professional and the trainee are displayed side-by-side, as shown in Figure 17, from the same camera 1310 at frame<sub>1</sub>. Determination of the first frame is described in greater detail below.

After the default view is displayed, the user may begin navigating the stored images. As with the embodiments described above, the user enters a user input via the user input device, and the server receives and interprets the input in a manner as described above with reference to Figures 5 and 6 (step 1615). The server then accesses and updates in parallel the trainee image (step 1620a) and the professional image (step 1620b).

In the present embodiment, the user inputs include moving to the left or right and up or down in the array; further, each directional movement can be forward in time, at the same point in time, or backward in time. Such movement is achieved by accessing and, where appropriate, stringing together the frames taken by the cameras. More specifically, navigating through the array can be based on the logical arrangement and addressing scheme of frames: to move to the left to the next camera 1310, the third digit of the address of the image to be viewed is incremented; to move up to the next row, the fifth digit of the address is incremented; to move forward in time to the next frame, the first digit of the address is incremented.

Thus, with reference to Figure 15, starting with the first frame--frame<sub>1</sub>--of row<sub>1</sub>(1) (i.e., the image stored at address 01 01 01) and moving to the left with the image frozen at the same point in time, the next image is that associated with frame<sub>1</sub> of row<sub>1</sub>(2) (i.e., the image stored at address 01 02 01), and then the image associated with frame<sub>1</sub> of row<sub>1</sub>(3) (i.e., the image stored at address 01 03 01). Similarly, starting with the first image -- frame<sub>1</sub> -- of row<sub>1</sub>(1) (i.e., the image stored at address 01 01 01) and moving up, to the left and forward in time, the next image could be that associated with frame<sub>2</sub> of row<sub>2</sub>(2) (i.e., the image stored at address 02 02 02), and then the image associated with frame<sub>3</sub> of row<sub>3</sub>(3) (i.e., the image stored at address 03 03 03). In embodiments where the images are stored as video streams, the viewer accesses the desired frame or time perspective of the images based on the synchronization of the image streams and/or time codes embedded in the streams.

Once the new camera outputs are accessed and retrieved from the storage devices, the server provides an updated view to the user (step 1625). Images of both the professional and trainee are updated synchronously. Changes to the user's view is applied to both the

professional's and the trainee's images. Operation of the present embodiment is made efficient by using the same addressing scheme in both the storage device containing the professional's images and the storage device containing the trainee's images. In other words, each frame from each camera is stored at the same address in different storage devices. Therefore, the server receives the user input, determines the next appropriate camera frame/output and corresponding address, mixes the last frame with the updated frame and causes the image stored at that address in each storage device to be provided to the user. Having displayed the view, the server awaits the next user input (step 1615).

In the embodiment of Figures 13-17, the server continuously updates the view based on the previously entered user input until the user enters a different input. Moreover, the playback preferably occurs at the same rate as the image capture occurred, namely thirty frames per second in the present embodiment. Therefore, when the selected user input is "forward in time" (from any camera(s)), the view is essentially a video playback at the actual speed of the swings. It should be understood that the present invention is independent of the type of cameras and the capture and playback rates.

The present embodiment thus allows for enhanced comparison of images and, consequently, improved training. The trainee's swing can be compared to that of the professional in many ways. For example, the swings can be compared at a single point in time, such as at the top of the trainee's back swing, and from any perspective provided by the array, such as front, back, top, etc. Additionally, the swings can be compared through sequential points in time, throughout a portion or the entirety of the swings, and from a changing perspective. The swings can be compared at actual speed over and over again, each

time from a new perspective. In sum, the present embodiment allows two images to be compared at any point in time from any perspective.

In an alternate embodiment for comparing multiple images, the images are displayed one overlaid on top of another. In one alternate embodiment utilizing overlaid images, the images are displayed with differing luminance levels. For example, the professional swing image, which remains constant, can be captured and stored with no change in luminance level. The trainee swing image, on the other hand, can be stored with a lesser luminance level so that it can be overlaid on top of the professional swing image. In such an embodiment, the camera outputs are temporarily stored in the storage device and retrieved by the server. The server not only processes the outputs to matte out the image (if desired), but also adjusts the luminance level of each image. The server then stores the processed outputs for later retrieval during playback. In related embodiments the luminance levels are adjusted at different points during the system operation, such as when originally retrieved from the cameras or just prior to outputting to the user interface display device.

In yet another alternate embodiment, the user may separately control the views of the professional's and the trainee's swings. In such an embodiment, the server discriminates between two sets of user inputs—one relating to each of the two images.

In the embodiment of Figures 13-17, the opening in the dome allows the golfers to take a realistic swing and hit an actual ball. Where a greater range of viewing is desired, however, the array need not include an opening for the ball to travel. Instead, the golfers can be completely enclosed in a dome of cameras (entering by way of a door having cameras mounted thereon), thereby allowing viewing from 360°.

In the embodiment of Figures 13-17, the server mixes the camera frames/images by electronically switching between frames/images. However, in alternate embodiments the server mixes the frames/images in any of the manners described above. For example, in one embodiment, mixing includes creating a "tweened" image from the output of adjacent cameras. The tweened image can be created and stored, or depending upon available processing power, created in real time as the view is being presented to the user.

Figure 18a illustrates the logical relationship of real and mixed images according to one embodiment in which the mixed images are synthesized images that are the product of images (output) from adjacent cameras. The logical arrangement of frames containing the real and mixed images can best be illustrated in the three dimensional representation in which the first access represents sequential frames, the second access represents sequential rows, and the third access represents sequential cameras in each row. Thus, as shown in Figure 18a, sequential frames of the same camera are illustrated along the horizontal axis (i.e., left to right), adjacent rows are illustrated along the vertical access, and adjacent cameras in the same row are illustrated along the access extending into the page. More specifically, frames containing real images are illustrated as squares and bear the same logical address as corresponding frames identified in Figure 15. Synthesized frames created by mixing outputs from the same point in time, from two adjacent cameras, in the same row are represented by triangles; synthesized frames created by mixing outputs from the same point in time, from corresponding cameras in adjacent rows are indicated by circles; and synthesized frames created by mixing outputs from the same point in time, from a camera in a given row and from the next camera in an adjacent row are indicated by diamonds. The asterisk indicates a

synthesized frame created by mixing the outputs from adjacent cameras, in adjacent rows taken at subsequent points in time (i.e., adjacent frames).

Furthermore, the mixed images are labeled with the logical notation wherein an apostrophe ('') adjacent to either the second or third pair of digits signifies that the image was created by mixing outputs of adjacent cameras in the same row or corresponding cameras in adjacent rows, respectively. For example, the notation 01'01 01 refers to the image created by mixing frames from 01 01 01 and 02 01 01; 01'01'01 refers to the image created by mixing the frames 01 01 01 and 02 02 01; and 01'01'01' refers to the image created by mixing the frames 01 01 01 and 02 02 02. It is to be understood that certain of the mixed frames, although described as being the product of two particular frames, may be the product of two or more other frames. For example, frame 01' 01' 01 may be created by mixing frames 02 01 01 and 01 02 01, or by mixing 01 01 01, 02 01 01, 01 02 01 and 02 02 01.

Although for simplicity Figure 18a illustrates only two successive frames of each of two adjacent cameras in each of two adjacent rows, it is to be understood that the logical depiction is readily extensible to multiple frames, cameras and rows. Having described the logical relationship of frames containing real images and synthesized frames containing mixed images, exemplary user navigation will be described with reference to Figures 18b and c, which use the same notation as Figure 18a, and continuing reference to Figure 13.

Thus, a user navigating the array from the first camera in row 1 and moving to the left at the same point in time is sequentially provided the images of frames 01 01 01, 01 01'01, and 01 02 01. Continuing to navigate the array by moving upward at the same point in time, the user is sequentially provided frames 01'02 01 and 02 02 01. Finally, moving forward in

time from the same camera the user is sequentially provided the image of frame 02 02 02 and subsequent frames, 02 02 03, 02 02 04, et seq.

Similarly, as shown in Figure 18c, a user navigating through the array diagonally to the left and up while moving forward in time is sequentially provided frames 01 01 01, 01'01'01' and 02 02 02.

In certain embodiments of the present invention the system identifies one or more reference points of the swings and uses such reference points to synchronize the swings and/or adjust the playback speed of the swings. In such embodiments, the system includes a user interface device, through which a user can manually indicate a reference point of a swing, or any number of motion measuring devices, such as motion detectors, range finders, electronic tags (mounted on the golfer or golf club) and the like. Applying such devices to embodiments of the present invention, various points in the swing can be identified, including the beginning of movement of the golf club during the back swing, the change of direction of the golf club at the end of the back swing, contact of the golf club and the golf ball, the end of the follow-through, when the golf club comes to rest, and the like. Manual indications, as well as indications received from such movement measuring means, of the various points in the swing may be used to synchronize the swings of the professional and the trainee.

In such an embodiment, the system begins recording of a swing at a reference time,  $t=0$ . The system then receives an indication, either manual or from one of the motion measuring means, indicating the reference point in the swing. More specifically, the system automatically notes the time of such an indication,  $t=x$ , relative to the beginning of recording.

Having an indication of the time ( $t=x$ ) at which the reference point of the swing occurred, the system identifies the frame corresponding to the reference point essentially by multiplying the time at which the reference point of the swing occurred by the recording speed of the cameras (i.e.,  $x$  seconds (30 frames/second)= $30x$  frames). In an alternate embodiment, the system receives the indication and, in essentially real time, tags the corresponding reference frame.

With this process repeated for both the professional swing and the trainee swing, the two identified reference frames are used as synchronizing points for the swings. For example, in one embodiment where the reference point is the beginning of the back swing, such reference frames are used as the first frame in the playback and all navigation is performed relative to the two reference frames.

In the embodiment where the beginnings of the swings are synchronized, a user is able to compare the swings to determine whether the trainee is swinging too fast or too slow. However, where the trainee and professional swing at different speeds, point-by-point comparison of the swings becomes difficult as the swings diverge and lack synchronization. Use of multiple reference points, however, permit this system to synchronize the swings and compensate for the different swing speeds, thereby allowing essentially point-by-point comparison of the swings.

The operation of one embodiment in which the system uses multiple reference points and compensates for differing speed swings will now be described with reference to Figures 19-20. Different swing speeds correspond to different time durations of swings, which in turn, correspond to different number of frames. Thus, one manner in which to compensate for different swing speeds is to adjust the number of frames for one of the swings.

For example, a professional golfer may swing in about two seconds. During the two-second swing, cameras operating at thirty frames per second capture sixty frames. A trainee, on the other hand, may swing slower, over three seconds. Thus, the trainee swing will take a total of ninety frames. Accordingly, with the playback of the images occurring at the same thirty-frames per second rate, the addition of thirty frames to the professional swing will cause the professional's swing to be the same duration and thus speed as the trainee's swing; both will be ninety frames in duration. In the present embodiment, these thirty additional frames are preferably mixed images created from successive frames of each camera that are uniformly interspersed among the frames of each camera containing real images.

The logical arrangement of the frames containing real images and frames containing mixed images of the foregoing example is illustrated in Figure 19. Interspersed among the sixty frames containing real images of the professional swing are thirty frames of mixed images. More specifically, the thirty mixed images are uniformly interspersed between every other pair of frames; a mixed image has been created between frames 1 and 2, not between frames 2 and 3, between frames 3 and 4, not between frames 4 and 5, and so forth.

It is to be understood that such mixed images created from successive frames from the same camera can be combined in the same embodiment as mixed images created from frames from different cameras. Moreover, in certain embodiments of the present invention, such mixed images interspersed for the purpose of adjusting the speed of the image are used to create other mixed images. For example, in the schematic of Figure 18a, the mixed images that are interspersed for adjusting the speed of the swing are indicated by an "X", and (using the notation of Figure 18a) mixed images 01 01 01' and 02 01 01' are used to create mixed image 01' 01 01'.

The capture of the images and creation of the mixed images of the embodiment of Figure 19 will now be described with regard to Figure 20. The system first captures and stores the image of the professional's swing and the image of the trainee's swing (step 2010). The system then receives a user input via a user interface device indicating the user's desire to harmonize the speeds of two swings (step 2020). The system then proceeds to create the necessary mixed images.

More specifically, during playback of each image, the system receives indications via the motion measuring device coupled to the system (e.g., server) noting both the beginning and end of the first swing (step 2030). These user indications correspond to particular points in time relative to the start of recording, which, in turn, correspond to particular reference frames that the system tags. In alternate embodiments the system automatically identifies the beginning and end of each swing by input from any of a number of motion measuring devices, such as motion detectors, range finders, electronic tags and the like, and in other embodiments via manual input via a user interface device during playback of the images.

It should be noted that the "beginning" and "end" points of a swing need not be precisely defined, but are preferably selected so that the points correspond to the same part of the two swings. For example, the beginning may be the beginning of the golfer's back swing and the end may be when the golf club comes to rest after the golfer's follow-through.

Once the system has identified the bounds (i.e., beginning and end) of what the user considers to be the swing, the system determines the number of frames in the first swing (step 2040). In the present embodiment, the system determines the number of frames by noting the relative time between reference points and by multiplying by the number of frames

per unit time (e.g., x seconds) (30 frames/second) = 30x frames). In an alternate embodiment, the system determines the number of frames by incrementing a counter for each frame address in a linked list of frame address between the frames corresponding to the beginning and end of the swing. The system proceeds through the same steps to count the number of frames for the second swing (step 2050).

Once the number of frames in each swing containing real images is determined, the number of frames in the faster swing is subtracted from the number of frames in the slower swing, resulting in the number of mixed images to be added to the faster swing (step 2060). In the example of Figure 19, because the slower swing included ninety frames and the faster swing sixty frames, thirty mixed frames must be added to the faster swing.

The system must also determine the composition of the mixed images (step 2070). In the context of the logical depiction of Figure 19, the system must determine the "location" of the mixed images. Preferably, the system evenly intersperses the frames containing the mixed images. In the present embodiment, the location of the frames is determined by dividing the number of additional mixed images to be added into the number of frames containing real images of the faster swing. In the example of Figure 19, then, sixty original frames divided by thirty additional mixed images, equals one added mixed image every two original frames. Where the division results in a non-integer, even distribution can be approximated by rounding the result to the next highest integer. Each mixed image comprises the product of mixing the two adjacent frames containing real images.

Once the compositions of the mixed images are determined, the system proceeds to create and store the mixed images (step 2080).

It is to be understood that the present invention includes other manners of harmonizing the speed of the two swings. For example, in alternate embodiments rather than interleaving mixed images into the faster swing, blank frames are inserted or repeat frames are inserted. In still other alternate embodiments, the system accounts for the different speeds by adjusting the playback speed based on the ratio of the lengths of the swings. For example, in the context of the example of Figure 19, the playback speed of the professional swing (sixty frames) to trainee swing (ninety frames) is two-thirds (60 frames/90 frames) that of the trainee. Thus, if the trainee swing is played back at thirty frames per second, the professional swing is played back at twenty frames per second, resulting in swings lasting three seconds (60 frames (1 second/20 frames)=3 seconds; 90 frames (1 second/30 frames)=3 seconds). The system adjusts the playback speed by accessing and/or refreshing the frames at different rates. In yet another alternate embodiment, a number of frames (equal to the number otherwise to be added to the faster swing in the above embodiments) from the slower swing are dropped from the image.

Moreover, it is to be understood that the system and method for adjusting the speed of a swing may be separately applied to portions of a swing, thereby synchronizing discrete portions of swings. For example, the different durations of the professional's and trainee's backswings may be harmonized so that upon playback both images arrive at the end of the backswing at the same time. Furthermore, the remainder of the swing (i.e., the downswing and follow-through) can similarly be synchronized. To achieve synchronization of portions of the swing, the process of Figure 20 is performed based on the beginning and end of each portion of the swing to be synchronized.

Although certain logical storage arrangements of frames have been described herein, it is to be understood that the present invention is not limited to any particular frame addressing scheme. One exemplary addressing scheme is that of the embodiment of Figure 15, wherein successive images are stored at known, continuous addresses. In alternate embodiments, the system includes various degrees of a linked list of frame addresses.

In one such embodiment, each data element in the linked list points to a frame as well as the previous and successive frame in each of the variable dimensions, such as those illustrated in Figure 18a, including up and down, diagonal, left and right and forward and back in time. In other such embodiments, the data elements in the linked list point to either the previous or successive frame in a subset of those dimensions. Furthermore, it is preferable that frames taken from cameras at the boundaries of the array are linked to frames taken at the opposite boundary. For example, the frames from the last camera in a given row of the array of Figure 13 are linked to frames from the first camera in the same row.

Additionally, although the exemplary embodiments described herein relating to harmonizing the speed and duration of images are concerned with harmonizing two images, the present invention can be used to harmonize multiple images by utilizing the process described with reference to Figure 19 to add frames to all but the longest image. Furthermore, it is to be understood that although the embodiments described herein intersperse a single frame containing a mixed image between frames containing real images, in alternate embodiments multiple frames containing mixed images are interspersed between frames containing real images.

It is also to be understood that images captured and processed according to the present invention may be stored on a portable storage medium, such as a CD-ROM, and

played back by a user on hardware separate from that which was used to capture and process the images. In such an embodiment, the playback hardware includes software providing the play back functionality, including the ability to interpret user inputs and, in response thereto, locate and display appropriate frames. The playback software locates the frames in any number of ways, including accessing a mapping or linked list of the frames which is stored on the storage medium.

#### **10. matrix Viewer Overview**

The user interface devices of the foregoing embodiments may be thought of as a matrix player, serving as the interface to the user to manipulate and display the matrix of real and artificial (i.e., synthesized) video captured over time from any one of the foregoing embodiments. The matrix viewer allows the user to navigate through a path over the camera array (and the matrix of real and artificial frames) and to generate continuous video along this path. In short, the primary goal is for the viewer to transition through the views in a smooth and intuitive fashion.

As noted above, the matrix viewer performs different levels of image processing in different embodiments. For example, in certain embodiments where the user navigates both images captured from cameras (i.e., "real" images) and artificial images from perspectives in-between the camera images (i.e., "tweened" images), the matrix viewer receives both the real image streams and the tweened image streams, the tweened image streams preferably having been previously generated by another processing device of the system. In other embodiments, the matrix viewer receives the real image streams and information necessary to generate the tweened images.

One embodiment in which the matrix viewer receives real images and information necessary to create artificial tweened images will now be described in greater detail. The matrix viewer includes a rendering engine software component that uses pre-computed flow fields between camera positions to synthesize one or more in-between viewpoints. The method for view synthesis consists of a flow based warping step and image fusion step, as described more fully below. Distance to the synthesized view relative to the distance between the two viewpoints is used as a weight to compute the flow field to the new view. These flow fields are then used to warp the video onto the new view's frame of reference. These warped frames can be then combined in many ways. If the system uses only two views (camera inputs) to create the in-between or tweened view, a weighted average (where the weights correspond to the relative distances from the original frames) of the warped frames yield good results. If the system uses multiple views to created the tweened view, trimmed mean, least median squares and other robust statistical median methods may be used. This allows some robustness against outliers (such as ones produced due to bad flow fields). In an alternate embodiments, it is also possible to use pyramid based fusion methods to combine these images.

The methods used in the warping step can lead to different levels of complexity and speed at runtime. In one embodiment, a nearest neighbor interpolation of the flow fields yields relatively quick rendering results. In another embodiment, results can be obtained by first doubling the flow field resolutions before a nearest neighbor interpolation is performed. In embodiments where there is enough computing power available (to the matrix viewer if not preprocessed by another device), bilinear or even higher order interpolates can be used.

Flow fields may contain inaccuracies at occlusion and image boundaries. Based on forward and backward flow consistency and based on correlation measures between a frame and its warped counterpart, mask images of reliable flow areas may be generated. The rendering engine can then use these masks to decide which areas of each video frame contribute towards the final view presented by the viewer. It should be understood, however, that in many applications certain levels of inaccuracies are acceptable.

An exemplary design schematic of a matrix viewer is shown in Figure 21. In the present embodiment, the matrix viewer separates the user interface from the rendering or processing engine. The primary software component of the rendering system, the "matrix server," receives user input or guidance (movement) requests and render requests from the main software component providing the graphical user interface (GUI). In response to user inputs, the matrix server returns a real or synthesized view. The navigation controls are tied to a keyboard or a joystick, and the GUI sends these as directional and velocity request to the Viewpoint Controller. The Viewpoint Controller buffers these requests and pre-fetches the data (as described below) as needed in preparation for a request from the Callback Renderer. The GUI requests new views through a callback function to the Callback Renderer. Ideally this callback is made from a timer loop within the GUI. This enables rendering at a fixed frame rate. In response to such a call, the Callback Renderer requests the position information and the corresponding data from the Viewpoint Controller and generates the new view.

The matrix viewer of the present embodiment preferably runs on a user interface/display device including at least the processing power of a Pentium III Class Windows NT machine with 256MB of Memory; however, other computers, operating

systems and processing devices are suitable. The system preferably includes a high-speed disk that can accommodate the multiple video streams and the flow fields that are generated. It is also preferable to have a graphics card capable of blitting the rendered video frames. It is further preferred that a dual processor system be used with multi-threading available with the matrix Server.

The operation of the matrix viewer will now be described in greater detail in the context of a particular example in which real images have been captured from a linear array of cameras arranged in an arc of, for example, approximately 120 degrees with a radius of approximately 15 feet. As such, in the present example, a linear camera array is used, thereby restricting tweening to horizontal neighbors for illustrative purposes. Although in alternate embodiments tweening may be in two or more planes or directions, depending upon the array configuration. Furthermore, the system of the present embodiment includes multiple personal computers (PCs). In addition to one PC reserved for performing controlling functions, each of the other PCs is utilized for capturing and processing real images. More specifically, each such capture PC includes two video capture cards or "frame grabbers," such as those provided by the Matrox Corporation under the tradename METEOR, captures the video stream from two video cameras. The controller PC performs controlling functions, such as driving and synchronizing the operation of the capture PCs. For ease of reference, the sequence of cameras in the linear array are logically numbered from left to right. The exemplary system configuration is illustrated in Fig. 22.

Once the system captures the real images (as described above), the capture PCs proceed to perform image processing on the real image streams. Such processing will now be described with reference to Fig. 23. In general, the present embodiment uses a nearest

neighbor interpolation of the flow fields to generate new views. Only two adjacent camera nodes are used at a time for tweening, and these nodes are limited to horizontal neighbors. In the present embodiment, the flow fields are pre-computed, preprocessed and provided to the matrix viewer so that the matrix viewer synthesizes new artificial views "on-the-fly." (See Fig. 23). In alternate embodiments where the matrix viewer has sufficient processing power, both real and artificial views are pre-generated, and the matrix viewer presents user-selected views. (See Fig. 24). In other alternate embodiments, the matrix viewer generates the flow field data and artificial images necessary to present the user-selected views. Playback can be set for any of a number of frames/second values, such as 15 frames/second.

It is to be understood that to maintain speed and quality, viewpoints (i.e., real or artificial perspectives) can be skipped to keep a realistic continuous motion in response to the user navigation controls. Although not required, the matrix viewer may also utilize a mask with the flow field to create a better synthesis of the views. The system may also use the Katmai (floating point) instruction set (preferably, on Pentium III or better systems) to improve the performance of the flow field interpolation.

Once the video is captured, each capture PC preferably saves the digitized images to disk (referred to in Fig. 23 as ".KMF" file) and converts each frame into bitmap (BMP) format file. The result is a series of BMP files for each camera real image stream. Furthermore, the system provides seamless navigation not only between each pair of cameras associated with a given capture PC (e.g., cameras 1 and 2), but also between adjacent cameras associated with different capture PCs (e.g., cameras 2 and 3). Consequently, each capture PC (other than the PC associated with the last two cameras) provides to the capture PC associated with the next two cameras the BMP files associated with the BMP files

associated with the adjacent camera. For example, where the first capture PC is associated with cameras 1 and 2, and the second capture PC is associated with cameras 3 and 4, the first capture PC provides the second capture PC with the BMP files representing the real image stream from camera 2. As such, each capture PC has three series of BMP files. As will be understood by those skilled in the art, in embodiments utilizing a circular camera array, the real images from the last camera must be mixed with the real image from the first camera to provide an arc or tweening path completely around the circular array. Accordingly, the BMP files associated with the last camera will be copied to the capture PC associated with the first camera.

Each capture PC proceeds to calculate both forward and reverse flow field data for each frame in the series of BMP files. More specifically, the capture PC calculates forward and reverse flow fields between the BMP files received from the adjacent capture PC and the BMP files associated with the first camera coupled to this capture PC and between the BMP files associated with the first camera coupled to this capture PC and the BMP files associated with the second camera associated with this capture PC. The flow fields are performed on a frame-by-frame (or BMP file-by-BMP file) basis for all frames (or BMP files). In the present embodiment, the cameras are synchronized during capture and the flow fields are generated between frames taken at the same time instant. For example, the first frame of camera 1 and the first frame of camera 2 are used to generate flow field data. As used herein, forward flow field refers to the flow field from the lower numbered camera (i.e., left) to the higher numbered camera (i.e., right) and reverse flow fields refers to the flow field from the higher numbered camera to the lower numbered camera. Upon completion of this part of the process, each capture PC will contain seven series of files: two series of BMP files

representing the real image streams from two cameras associated with the capture PC; one series of BMP files received from an adjacent capture PC; two forward flow field data files and two reverse flow field data files. The flow field data files can be a sequence of BMP files.

The capture PCs preferably convert the BMP files and the flow field files into the same multimedia file format, such as Audio Video Interleaved (AVI) format established by the Microsoft Corporation or other video format. In the present embodiment, the BMP files are merged into AVI format files. The capture PCs perform the conversion in any of a number of ways, including using a distributed system of components, such as Distributed Component Object Model (DCOM). The real image AVI files and flow field AVI files are utilized by the matrix viewer to permit seamless navigation by the end user.

In the present embodiment, the software components comprising the matrix viewer are installed on one of the capture PCs. As such, all AVI files are transferred to the capture PC having the matrix viewer. In alternate embodiments, one or more matrix viewers reside on separate user interface or processing devices coupled to one or more capture PCs by any now known or hereafter known technologies and protocols, including the Internet, Local Area Networks, Wide Area Networks, wireless transmission, and the like.

Although not necessary in all embodiments of the present invention, the matrix viewer according to the present example also utilizes a camera graph file (referred to as the kwz file in the figures) to permit navigation through the images by an end user. The camera graph file is loaded once by the matrix viewer at startup and informs the matrix viewer of the layout of the cameras in the array. More specifically, the file includes "nodes", which indicate actual camera positions, and "arcs", which indicate navigation paths between nodes

or cameras. As shown in the following exemplary file format (where the bracketed text is a comment), the camera graph file includes various other information.

Path = "string" [where string is the prefix path for all AVI or other video files]

FrameRate = <integer> [the integer number of the frame rate override]

Nodes = <integer> [the integer number of camera nodes]

Node {

X = <integer> [horizontal (X) coordinate of camera position in array, counting each tweened image as a node]

Y = <integer> [vertical (Y) coordinate of camera position in the array, in present example with linear array, always equal to 0]

File = "string" [where string is the name of AVI stream for the node]

}

[in actual file, coordinates and file name for each remaining node provided]

Arcs = <integer> [number of arcs in array; e.g., in a linear array of six cameras there are five arcs]

Arc {

Node 1 = <integer> [number of first index node; i.e., node defining beginning of arc]

Node 2 = <integer> [number of second index node; i.e., node defining end of arc]

FileF = "string" [where string is the name of file containing forward flow data for arc]

FileR = "string" [where string is the name of file containing reverse flow data for arc]

}

[in actual file, index nodes and file names provided for each remaining arc]

In alternate embodiments, instead of pointing to an AVI filename, one or more nodes may include a reference to another node and, consequently, to the file associated with that

other node. The AVI stream of the referred-to node is segmented equally among the referred-to node and the referring nodes. Furthermore, each such referring node also includes an indication of the starting frame of the corresponding segment of the stream. For example, if the referred-to node has an AVI file of 1000 frames and there are nine nodes referring to the node, the each node will have an AVI segment of 100 frames (1000 divided by (9+1)). Similarly, an arc can reference another arc flow field files so that the flow field file can be segmented. The starting frame of the flow field file is also similarly specified.

Once the system has generated the real image streams, the flow field data files and the camera graph and has made them available to the matrix viewer, the matrix viewer proceeds to retrieve real images and generate artificial images in response to user inputs.

The processing of artificial images (in the present embodiment, tweened images) will now be described in greater detail. As an initial matter, however, it should be noted that the artificial images may be created in any of heretofore or hereafter known manners. When creating artificial or tweened images, the matrix viewer computes correspondences between the images acquired from one camera with the images acquired from the other cameras. The correspondences are preferably computed between frames from adjacent views acquired at the same time instant. However, in alternate embodiments correspondences may be computed between frames at different time instants or non-adjacent cameras. (See, e.g., Fig.

18) Having performed the correspondence, the matrix viewer transforms the correspondence mappings such that the mappings point to the desired virtual viewpoint. The desired virtual viewpoint is determined based on the number of virtual camera perspectives between each pair of real cameras (as identified in the camera graph file) and interpolating to the virtual position of each virtual camera perspective. With the mappings to the desired virtual

viewpoint determined, the matrix viewer warps or shifts the pixels in each image using the transformed mappings so that all pixels are in the coordinate system of the desired synthesized viewpoint.

In the present embodiment, tweening involves the computation of correspondences between images acquired from different camera views. Dense correspondences are preferred throughout the image since it is desired to display a complete image, and not just recover pose or some other information that typically requires fewer correspondences, although the density of correspondences often depends on the application. Correspondences are generated between at least two, preferably adjacent, views or cameras. However there are advantages in resolving occlusion problems if correspondences are computed between more than two views. As will be understood by one skilled in the art, there are several problems that can occur in the computation of correspondences. These include gain and offset variation between cameras, the aperture problem where only a portion of an image feature (a straight line for example) is visible so that there is insufficient information locally to provide complete correspondence information, textureless areas where there is no local image information at all to provide correspondence information, and occlusion and dis-occlusion of areas such that corresponding areas do not exist between two images. Camera resolution also does have some influence in the correspondence process in the sense that some parts of the scene can only be matched above a certain resolution. There are several methods for performing correspondence. Each of these methods are now discussed.

One reliable and simple method of computing dense correspondence is coarse-to-fine flow algorithm over an image pyramid [15] (bracketed numbers refers to related references identified below, each of which is hereby incorporated herein by reference). This algorithm

uses an image matching constraint to locate corresponding local patterns between two images. More precisely, it uses the assumption that brightness is constant between two or more images, and solves a brightness constancy equation locally throughout the image to compute a correspondence. In order to overcome brightness variations due to the use of different camera sensors, the images are pre-processed using a Laplacian filter such that low-frequency gradients are removed. A first feature of this algorithm is successive refinement. In this process an initial correspondence is computed as described above, and then the images are warped using the result. Correspondences are then re-computed. This compute/warp process is repeated several times. This greatly improves the accuracy of the correspondences. A second feature of this algorithm is coarse-to-fine refinement. In this process, the compute/warp procedure is performed first at coarse image resolutions and then refined at finer scales. This improves the ability to deal with longer range motion, and also increases robustness in areas of the image that are textureless or contain aperture problems. The basic flow algorithm can be modified to enhance performance. These performance enhancements include the use of a sliding local window for the correspondence calculation, rather than the use of a fixed window. Differently shaped windows have also been explored [20] and may be used in alternate embodiments. This provides enhanced performance at occlusion boundaries.

Secondly, the flow computation can be seeded by a parametric alignment step where a global affine (or quadratic, projective) transform is computed between the frames. [4,7,10,15]. This brings the images into rough alignment and reduces the range of matching needed to be done by the optic flow computation for sub-pixel alignment. One primary feature of the flow method is that the only constraint exploited is image matching. There are no three-dimensional (3D) constraints imposed. One advantage of this method is that since

image matching is the primary constraint (other than smoothness) that is imposed, correspondence and the resulting tweened results are typically relatively good in typical textured areas. Performance at occlusion boundaries is also acceptable, given that the camera spacing is sufficiently close. This is because a neighborhood window is used in the flow computation so that dramatic errors in correspondence only occur if the size of the occluded or dis-occluded region is larger than the neighborhood window size. A second advantage is computational efficiency. The algorithm does not require complex 3D calculation and is therefore relatively fast. A third advantage is the fact that the flow process does not require any intensive camera calibration or setup procedures.

In plane and/then parallax methods, 3D shape in the scene is represented by 2 components. The first component is a real or virtual 3D planar parametric surface in the scene, while the remaining residual 3D shape is represented by a non-parametric surface [12]. For example, the scene of objects lying on a floor would be represented by a 3D planar parametric surface corresponding to the floor, and then by a residual non-parametric surface that is a direct function of the heights of the objects above the floor. In the "Plane and Parallax" method, both the parametric and non-parametric surfaces are computed simultaneously [12]. In the "Plane then Parallax" method, the surfaces are recovered sequentially. This method has several advantages and disadvantages over flow and other 3D recovery methods (discussed later). An advantage over other 3D recovery methods is that the initial computation of the planar surface brings features into closer correspondence resulting in more accurate calculation of the non-parametric surface. The advantage over the flow method is that 3D information is recovered and this can be used to resolve occlusion problems (discussed in the selection/merging section). In the plane and/or parallax method,

camera pose estimation preferably either has to be recovered during the computation, or has to be provided accurately to the algorithm. The recovery of pose in the algorithm may not be sufficiently robust to deal with general scenes. Pose could be recovered in a calibration step for every camera in the system, however lens distortion and other factors would also require modeling. The plane and parallax algorithm is also significantly more computationally intensive than the basic flow algorithm. This algorithm can also be extended to compute 3D shape using more than a pair of images, using the constraint that the shape is constant.

The ego-motion method for 3D-shape recovery [2,18] can also be used and is similar to the plane-and-parallax method, except a single depth map represents the 3D shape of the scene. The advantage is that it is simpler than the plane and/or parallax methods to combine results from several image pairs to resolve occlusion boundaries. However a disadvantage is that long-range non-parametric correspondence is performed, and this is slightly less accurate than the parametric and non-parametric methods used in the plane and/then parallax methods. This algorithm can also be extended to compute 3D shape using more than a pair of images, using the constraint that the shape is constant [17]. A related algorithm uses correlation rather than the brightness constraint to compute correspondence [16]. The correlation approach offers the ability to perform long-range correspondence without resorting to coarse scales in the pyramid, which may blur features excessively. It is to be understood that any of the above three sets of alignment methods for computing correspondences between frames as well as essentially any other heretofore [e.g., 1,5,13,14,19] or hereafter known methods may be used. However, for illustrative purposes, only the foregoing three methods are discussed in detail.

The methods for computing plane and parallax, and other 3D methods are less robust in computation as compared to flow computation, and require more calibration (such as lens distortion [8] and camera pose calculations) and setup procedures. This becomes a very significant procedure when large numbers of cameras are involved. The disadvantage of using flow is that 3D information on the location of occluding and dis-occluding areas is not exploited. Second, the flow-based methods allow the synthesis of new views from view-points which lie in straight lines between the captured views when using two cameras. In contrast, the 3D methods compute depth and allow the generation of new views from arbitrary view-points which need not lie on the straight line between the view-points. These are the primary advantages of the 3D methods. However to exploit the constraint effectively and robustly, several steps are required: camera calibration for all cameras, robust pose estimation between camera pairs, and robust selection and merging of data from each image based on 3D position. These steps are possible but often require robustification of existing algorithms.

Finally, image based correspondence methods may fail when there is a little or no texture in the scene and when there is a lot of occlusion present. Some of this is mitigated by using image pyramids and multiple images to do alignment etc.. However, there can still be cases when these methods fail. These errors can be dealt with in a variety of methods, including, for example: 1. Post production editing of correspondence maps in regions of error; The correspondence maps can be examined by an operator and with a simple editing tool corrected in areas of error: 2. Active sensing methods, which use either project textured light patterns (in non-visible spectrum) or detect 3D range, can be used to provide more information for computing the correspondence maps. The active sensing methods can be

used to acquire the background scene and/ or during imaging of the live event. Various change detection algorithms, such as those developed at Sarnoff Corporation [e.g. 13], may be used to detect foreground objects in the live scene. The final correspondence maps between frames of the live scene are computed by intelligently selecting between correspondences computed for frames imaging the background scene and the correspondences of the frames for the live scene computed using optic flow or one of the other methods.

The next step is transforming the correspondence mappings. The results of the algorithms described above can be used to compute a new mapping between each image and the synthetic view. If 3D information is available, then the approach is to compute an intermediate pose between two or more camera positions, and to compute the flow field produced by combining that pose and the depth map or 3D representation recovered at that camera position. If only 2D information is available, then the approach is to compute the mapping as a fraction of the flow field from one image to the next.

Pixels are then warped using bi-linear or bi-cubic warping methods so the pixels from the processed images are in the coordinate system of the synthetic viewpoint. If 3D information such as a depth map is available, then pixels will not be warped from locations where the depth map indicates that there is a dis-occlusion or occlusion. This area will be flagged so that a selection process can later choose the best intensity from other images to produce a result. In certain embodiments, however, limited occlusion is acceptable.

Selection and merging are then performed. The correspondence algorithms that have been described can be performed both forwards and backwards on the same image pair, and can also be performed between different image pairs, both temporally and spatially. These

correspondence results are preferably combined. Real images are warped to the synthetic or artificial view-point. These images can be combined by a variety of methods to create the new synthetic image [3,11]. If 3D depth is computed at each image, then the image with the nearest scene-point is used to create the new image. This prevents occluded regions from being rendered from the new viewpoint. If more than one image renders this scene point, then these images may be combined by an average, trimmed mean or median operations. One effective combination method is to combine both background and forward results on the same image pair. This reduces the visibility of artifacts in the resulting image. A second effective combination method is to measure the residual error after warping by each (original) flow field and to weight or discard those pixels derived from flow fields with significant local error. More specifically, an alignment quality mask can be used together with forward/backward computation of flow to select and combine the intensities appropriately.

The objective for the display step is to show the tweened images in real-time. However since latency is allowed in the system, it is not necessary to perform all proceeding steps in real-time. Thus, in certain embodiments, the flow computation can be computed in non-real-time, and some parts of any flow-field selection and merging procedure may be computed in non real-time. The flow field can also be quantized both spatially and in bit-depth in order to reduce the required IO bandwidth from disk (or other storage) into the display device. The image warping preferably occurs in real-time.

It should be understood that in alternate embodiments all AVI files need not be transferred to a single computer or user interface device having a matrix viewer installed thereon. In one such alternate embodiment, the files are streamed from associated capture

PC's over a network connection to the capture pc or other device containing the matrix viewer. In one such embodiment, multiple end users, each having a user interface device, are coupled to the capture system via a network, such as the Internet. Each end user interface device includes a camera graph file as well as the matrix viewer software. The AVI files consisting of the real image streams and the flow field data are streamed to each end user. More specifically, each end user interface device engages in two-way communication with a processing system for transferring the real image and flow field data (e.g., server 18, 804, 902) so that the processing system is aware of what camera view each end user is currently viewing. By tracking the current perspective being viewed by each end user, the processing system is able to provide each end user with a limited number of AVI streams that are most likely to be needed by each end user. For example, if a given end user is currently viewing the output from camera 3, the processing system provides the end user with the AVI files logically surrounding camera 3, namely the flow field data files associated with the arcs between cameras 3 and 2 and camera 3 and 4, as well as the AVI files representing the real images from cameras 2 and 4.

As will be appreciated by one skilled in the art, a greater or fewer number of AVI files surrounding the end user's current perspective may be provided depending upon available bandwidth. As the end user continues to navigate the array, the processing system can anticipate the user's navigation and continuously provide the necessary AVI files. In other words, the capture and processing system provides each end user with the necessary files within a window (e.g., a number of real and/or virtual camera positions in any one or more directions from the currently viewed). In the present embodiment, the center of the window is preferably the user's current view. In other embodiments, the streaming data

provided to the end user are weighted to the current direction of movement. For example, if the user is navigating to the left, the system provides streams/files associated with more cameras and/or flow fields to the left of the end user's current view than associated with cameras to the right. Furthermore, in such an embodiment, the matrix viewer preferably limits the speed of navigation among real and virtual perspectives so that fewer AVI files need to be provided to each end user.

In certain embodiments, the matrix viewer automatically causes the end user's view to stop on the perspective of the next real camera in the direction the end user was navigating. This feature not only provides the user with clear (real) images when the user is not traversing the array, but also allows the system to conserve processing power and to better anticipate and provide the necessary AVI files for the end user's further navigation. It also minimizes the perception by the user of any ambient artifacts, which may be less apparent while the user is navigating between cameras, but more apparent when his/her perspective motion path has come to rest. Thus, stopping on an artifact-free real camera position insures that any transient artifacts will be less persistent and less noticeable to the user.

To summarize the matrix viewer of the present embodiment, it allows for the controlling of time and viewpoint, and warping of imagery in response to user inputs. The input to the matrix viewer can be either a raw set of pre-synthesized image data, or a set of original image data together with a set of flow-fields. In both cases the matrix viewer allows the user to navigate the data both in space and in time, with the use of 2 slider controls, a single graphical control (e.g., the four-quadrant button described above) and the like. In the

present embodiment, the matrix viewer thus works generally according to the following four steps.

Before being read into the matrix viewer, the data is first organized. The simplest organization would be to take all of the data and to store them sequentially in one file. Where the Operating System (OS) used cannot read or write the data file in a timely fashion because it is too large (e.g., many viewpoints and/or long sequence), the data is preferably pre-organized. Since real-time playback of the data is preferred, the accessing and retrieval of data from disk is preferably optimized. Accordingly, in the present embodiment the data is split into several smaller files, and data from adjacent viewpoints is grouped together in the same file. By producing several smaller files, the problem with the OS file size limitation was solved. In certain embodiments, the data (e.g., AVI) files are stacked, with one file stored in a related file. Such stacked files may include, for example, different temporal periods of the same viewpoint. Similarly, in certain embodiments, image data associated with a number of contiguous viewpoints for a predetermined (relatively short) period of time are stored together. Thus, the input data is preferably organized in order to minimize the number of O/S files that remain open when the data is read, and also so that the next likely temporal image that is to be displayed is likely to be contained within the same file as the current desired view. The advantage of grouping data from adjacent viewpoints together is discussed in third step.

The second step generally involves mapping the desired view given by the User Interface, and the next likely desired-view, onto the source data. The user can control the desired view using the GUI, for example, two slider bars (one for spatial input and one for temporal input). The temporal slider bar is either manually controlled or computer controlled

such that it can play back a sequence automatically in time. The position slider bar is controlled manually. The positions of the two slider bars are fed to an indexing system, such as, for example, a file stored locally like the .kwz graphing file noted above. This maps the desired imagery to a file name and a byte offset in the file where either the imagery to be displayed is located (in the case when pre-synthesized imagery is used as data), or the raw imagery and the flow fields are located (when on-the-fly image synthesis is performed). In certain embodiments, the byte offset corresponds to the frame number or temporal position in the stream, as each frame in such embodiments contain the same amount of data. In alternate embodiments, other methods of determining the position within the file and appropriate frame may be used, such as an index. The mapping is performed not only for the desired frame, but also for the next expected frame in the temporal sequence. This is to allow prefetching to be performed, to optimize the speed performance of the viewer. The mapping function is architected to allow navigation both in the temporal axis, as well as, depending upon the configuration of the array, in one or more spatial dimensions.

The third step uses the mapping performed in the second step to fetch the data for the desired view and the next (anticipated) desired view into local memory. More specifically, the mapping performed in the second step is used to determine the file that should be opened (if it is not open already), and the (byte or other) offset from the beginning of the file from which data should be read. It is important to note that at least for a WINDOWS-based OS, it can take a relatively long time just to open a file, even without reading a byte of data into memory. As a result, it is advantageous to minimize the number of files that are required to be opened when a sequence is played. This can be done in the first step by grouping data from adjacent viewpoints into the same file.

The forth step involves processing the data for the desired view as required, displaying the view on the screen, and continuing to fetch data for the next desired view. If the data being read contained pre-synthesized imagery, then there is very little processing that is required to display the image data. The data is simply read from computer memory into the display memory. On the other hand, if the imagery has not been pre-synthesized, then the original image data and meta-data (in the present embodiments, flow fields were used) are processed by the processor/CPU to synthesize a new image using existing image synthesis algorithms. While the processor/CPU is performing its processing, the I/O module will continue to fetch data for the next desired view from memory. It is desirable to display imagery in real-time, which in this particular system corresponds to 30 frames per second. Thus, the viewer includes a timing function/module, which ensures that the update rate does not exceed this rate.

It is anticipated that embodiments of the present invention will be applied in the context of "replays" of sporting events. In such an application, however, it is not economical to capture and process every frame of video. Instead, it is preferable to capture video continuously and only process that portion which will be used for the replay. Such an embodiment preferably includes a GUI that allows a producer to control the acquisition, storage and processing of the video. In operation, the system continuously acquires video output from each camera, saving the video of each camera to a particular memory unit or drive. Once the memory is used up, the system overwrites the previously stored video in the same memory. Once a noteworthy event occurs, the producer enters an input causing the system to stop the process of overwriting the captured video, freezing a predetermined amount of video (e.g., fixed number of frames or seconds) within the memory.

The producer may then view the stored video and select all or a portion of the video for processing. Selection of the video portion to be processed is achieved through electronic tagging of the beginning frame and ending frame of the portion to be processed. In certain circumstances, both the beginning and end points may be a single frame. With the portion of the captured video to be processed identified, the system proceeds to process the video as discussed above, creating files representative of the real images, the flow field data, and eventually the artificial views.

The GUI for one embodiment of the matrix viewer consists of two slider bars. One bar controls the temporal advancement of the image sequence from whatever real or synthetic camera node it rests upon or is linked to. It moves the frame sequence forward or backward in time. In certain embodiments, the speed of playback is also selected by the end user. Such a temporal input adjusts the rate of the matrix viewer timer used to generate the images.

The other bar controls camera perspective, enabling movement from camera node to camera node, between the real and intervening synthetic camera positions, so that the user can fluidly guide his/her perspective around and through the viewpath (the aggregate available perspective motion path views within the camera array, both real and synthetic). Each location of the slider bar is mapped to a camera node. For example, the leftmost position of the slider bar corresponds to the leftmost camera, and the rightmost position of the slider bar corresponds to the rightmost camera. Spaced proportionally in between the two extremes are locations corresponding to each of the remaining real and artificial cameras/viewpoints. The matrix viewer maps or correlates the position of the slider bar to the desired real camera and its associated stream or a virtual camera and its associated stream

(which depending on the embodiment may need to be dynamically generated based on flow field data). The number of camera nodes can be pre-programmed into the viewer, or the camera graph file provides an indication of the number of camera nodes. Based on the timer noted above, the matrix viewer generates the desired image.

Another embodiment of the matrix player GUI integrates both the temporal and the viewpoint sliders into a single control enabling the user to navigate both dimensionally and in time using a single, less complicated, and less time-consuming gesture. In one embodiment, the GUI includes a grid having four quadrants: the top-left quadrant relates to movement forward in time and to the left in the array; the top right quadrant relates to movement forward in time and to the right in the array; the bottom right quadrant relates to movement back in time and to the right in the array; and the bottom left quadrant relates to movement back in time and to the left in the array. Other multi-sectioned or partitioned buttons could be used in which the sections correspond to a temporal movement and directional movement. The end user indicates movement through the array by placing a mouse cursor or other indicator in the grid: the further the indicator is from the center axis, the more pronounced the movement or navigation. More specifically, as the user moves the indicator further to the right, the user navigates faster through the array to the right. Likewise, movement of the indicator further to the left corresponds to faster movement to the left through the array. With regard to temporal aspects of navigating, the further the user moves the indicator to the top of the grid, the faster the image is played back, forward in time. Likewise, the further to the bottom the user moves the indicator, the faster the image is played back, back in time. It is to be understood that other arrangements in which the single location of an indicator

correlates into two playback variables (e.g., direction of navigation and temporal aspect of playback) may be used.

The GUI can also be with a joystick-like control, a position-sensor/control technology, or any wired or wireless mouse control function. By clicking on an ergonomically placed feature add/drop function button, it will also be able to combine or drop either of the slider parameters so that it can also function as a controller of either of the individual slider bars by themselves.

Other embodiments of the matrix player include a 2D or 3D graphic visualization map reference of the camera array(s) layout and placement within the physical environment in which it exists (or existed). One example is a map built on a substrate of pixel groupings that represent a field of all possible array shapes within a flat plane (overhead 2D) or an array of stacked planes (three-dimensional). The relevant camera positions within an array can then be activated or highlighted and scaled, based on real camera position data from position sensors in the real world camera environment. The user can click on a camera position within the map to view a particular camera perspective or slide a cursor along a highlighted path to fluidly guide his/her perspective motion path throughout the accessible viewpath. This viewpath can be within a single camera array or throughout a series of concentric, separated, or tangentially connecting arrays, either static or in movement.

It should be understood that the foregoing description of the matrix viewer also applies to embodiments utilizing only real images. In one such an embodiment, the user inputs (whether they be via a slide bar or other GUI) correspond to one of the real camera streams. Navigation among the streams is accomplished by jumping from a frame or time perspective in one stream to the desired frame or time perspective in a second, selected

stream, based on the user inputs. Such an embodiments can also utilize the feature described above, by which a fixed number of streams are provided to the end user based on the user's current viewing perspective(i.e., camera being viewed) and/or navigational direction.

Those skilled in the art will recognize that the method and system of the present invention has many applications, may be implemented in many manners and, as such, is not to be limited by the foregoing exemplary embodiments and examples. Moreover, the scope of the present invention covers conventionally known and future developed variations and modifications to the system components described herein, as would be understood by those skilled in the art.

**References:**

1. "Three-frame technique for analyzing two motions in successive image frames dynamically," US Patent 5,067,014; Issue date: 11/19/91.
2. "Method for determining sensor motion and scene structure and image processing system therefor," US Patent 5,259,040.
3. "Method for fusing images and apparatus therefor," US Patent 5,488,674; Issue date: 1/30/96.
4. "System and method for electronic image stabilization," US Patent 5,629,998; Issue date: 5/13/97.
5. "System for automatically aligning images to form a mosaic image," US Patent 5,649,032; Issue date: 7/15/97.
6. "Method for estimating the location of an image target," US Patent 5,581,629; Issue date 12/3/96.
7. "Method and apparatus for mosaic image construction," US Patent 6,075,905; Issue date: 6/13/00..
8. "Multi-view image registration with application to mosaicing and lens distortion correction," US Patent 6,173,087; Issue date: 1/09/01.
9. "Method and apparatus for processing image pyramid borders," US Patent 6,141,459; Issue date: 10/31/00.
10. "Method and apparatus for performing local to global multi-frame alignment to construct mosaic images." US Patent 6,078,701; Issue date: 6/20/00.
11. "Method and system for rendering and combining images," EP pub. application EP838068A1; Filing date: 07/08/96.

12. "Method and system for image combination using a parallax-based technique," US Patent 5,963,664; Issue date: 10/5/99.
13. "Method and apparatus for detecting moving objects in two-and three dimensional scenes" US Patent 6,049,619; Issue date: 4/11/00.
14. "Method and apparatus for three-dimensional scene processing using parallax geometry of pairs of points." US Patent 6,192,145; Issue date: 2/20/01.
15. "Hierarchical model based motion estimation," Bergen, Anandan, Hingorani and Hanna, Proceedings of the 2 nd European Conference on Computer Vision, Springer Verlag, pp. 237-252, 1992.
16. "Correlation-Based Estimation of Ego-motion and Structure from Motion and Stereo", R. Mandelbaum, G. Salgian, H. Sawhney, Proceedings of the International Conference on Computer Vision (ICCV'98), Kerkyra, Greece, September 1999.
17. "Combining stereo and motion analysis for direct estimation of scene structure," K. J. Hanna and Neil E. Okamoto, In Proceedings IEEE International Conference on Computer Vision, pages 357-265, 1993.
18. "Direct Multi-resolution estimation of ego-motion and structure from motion," K. J. Hanna, In Proceedings of the IEEE Workshop on Visual Motion, Pages 156-162, 1991.
19. "Accurate computation of optical flow by using layered motion motion representation," S. Hsu, P. Anandan, and S. Peleg," In Proceedings of the International Conference on Pattern Recognition, Jerusalem. October 1994.
20. "Method and apparatus for processing high resolution images." Pending US application filed 8/27/99.

**CLAIMS**

1. A method of displaying images captured from an array of cameras, the method comprising:

pre-organizing data representing the images;  
receiving an indication of a desired view;  
determining a next potential desired view;  
mapping the desired view and the next potential desired view onto the data;  
retrieving data associated with the desired view;  
retrieving data associated with the next potential desired view;  
processing the data associated with the desired view; and  
displaying the desired view.

2. The method of claim 1 wherein determining the next potential desired view is based at least in part on the desired view.

3. The method of claim 1 wherein determining the next potential desired view is based at least in part on a direction of movement through the array.

4. The method of claim 1 wherein pre-organizing the data includes organizing the data based on camera position.

5. The method of claim 1 wherein pre-organizing the data includes organizing data representing images taken at a particular time.

6. The method of claim 1 wherein mapping the desired view includes accessing a graph file.

7. The method of claim 1 wherein the desired view corresponds to a time and mapping the desired view includes calculating an offset corresponding to the time.

8. The method of claim 1 further comprising:

receiving an input corresponding to the next potential desired view;

processing the data associated with the next potential desired view; and

displaying the next potential desired view.

9. The method of claim 8 further comprising: repeatedly retrieving data associated with another potential desired view prior to receiving an indication to display the another desired view.

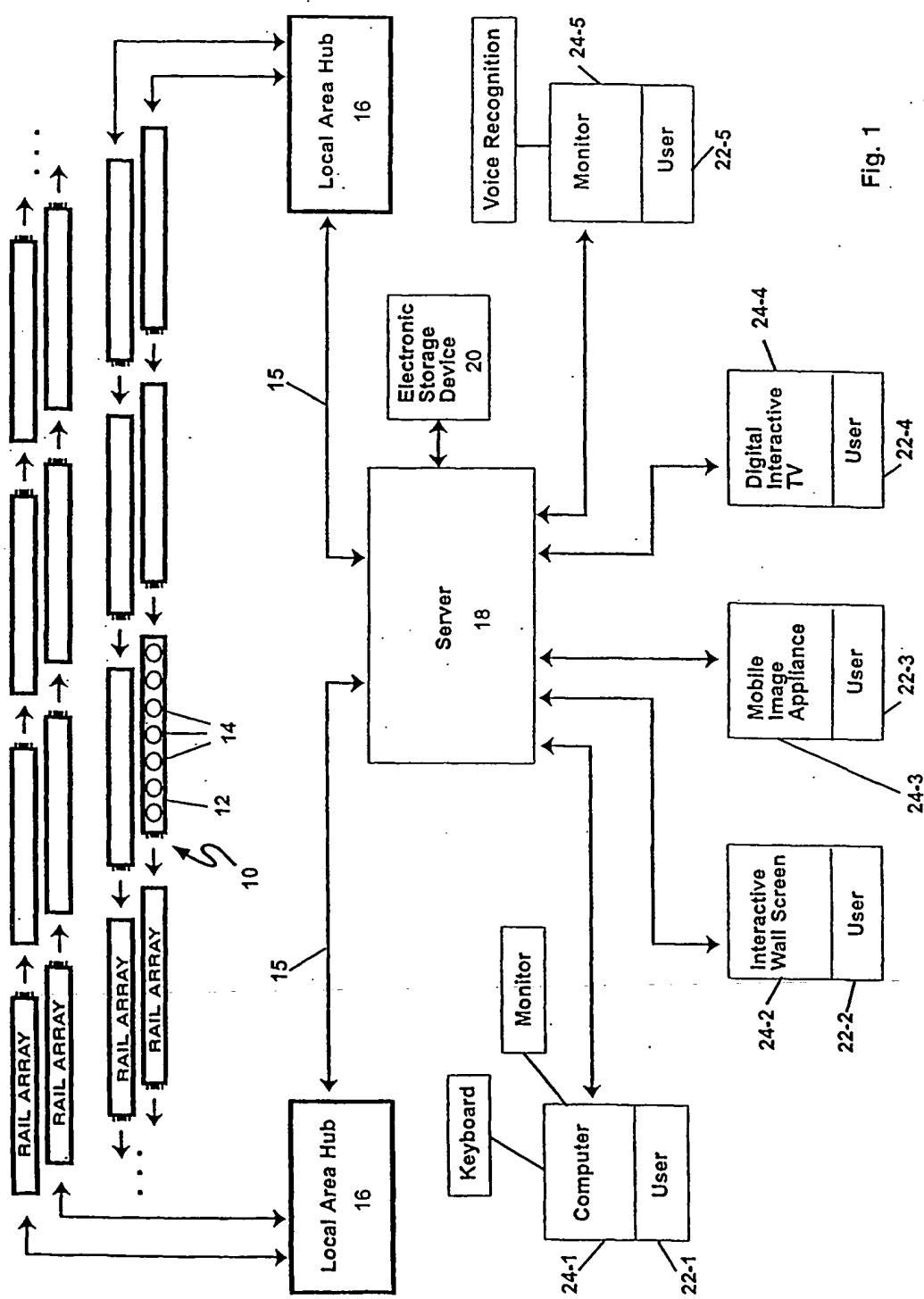


Fig. 1

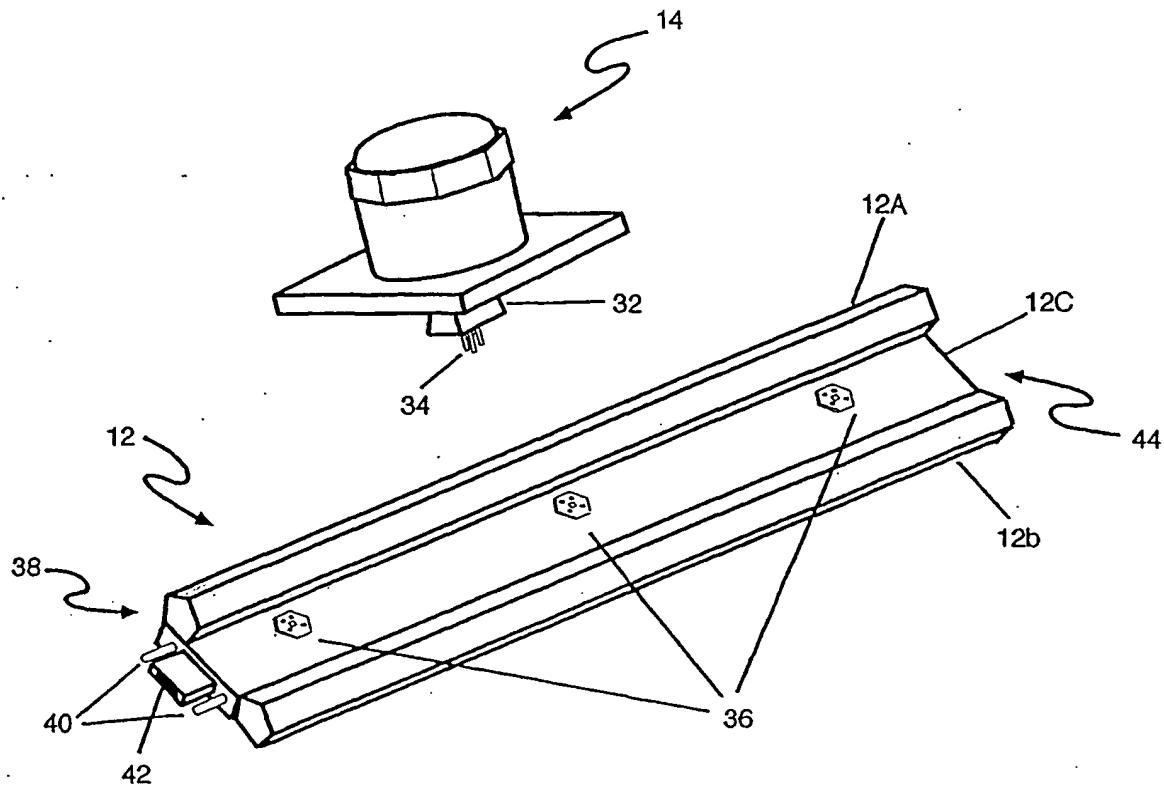


Fig.-2a

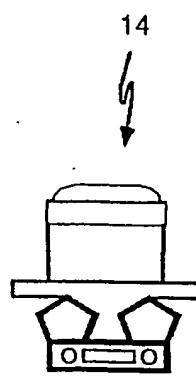
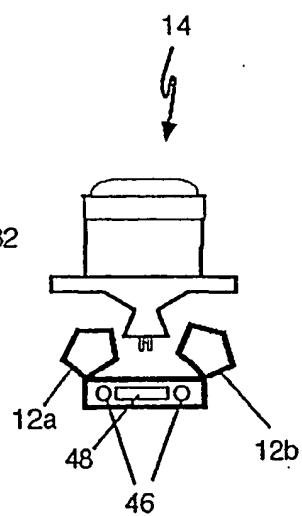
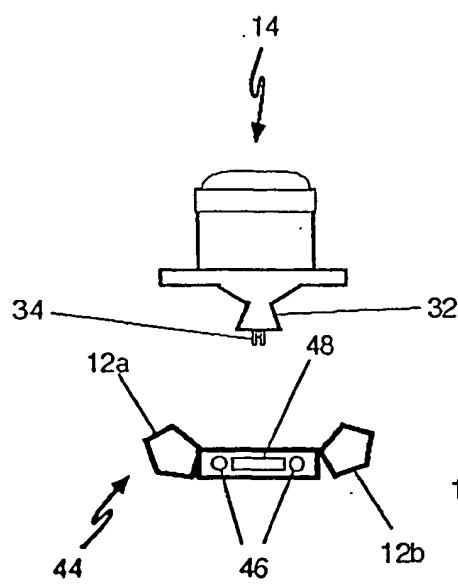


Fig. 2b

Fig. 2c

Fig. 2d

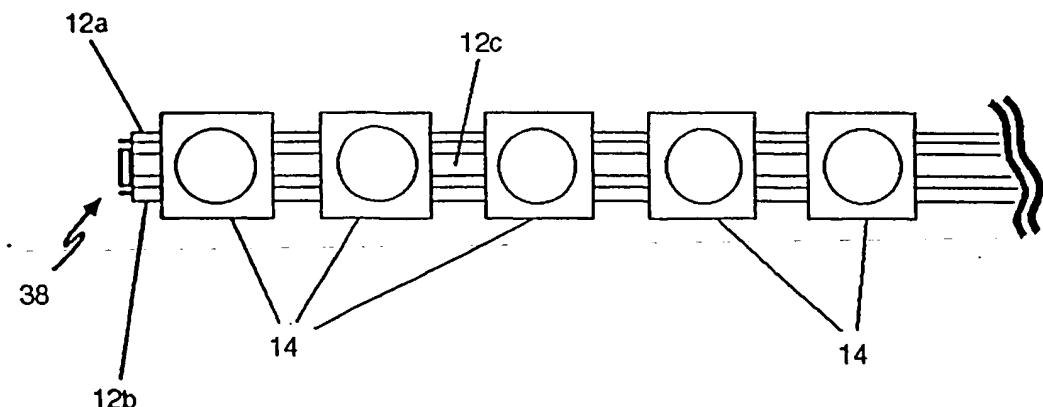


Fig. 2e

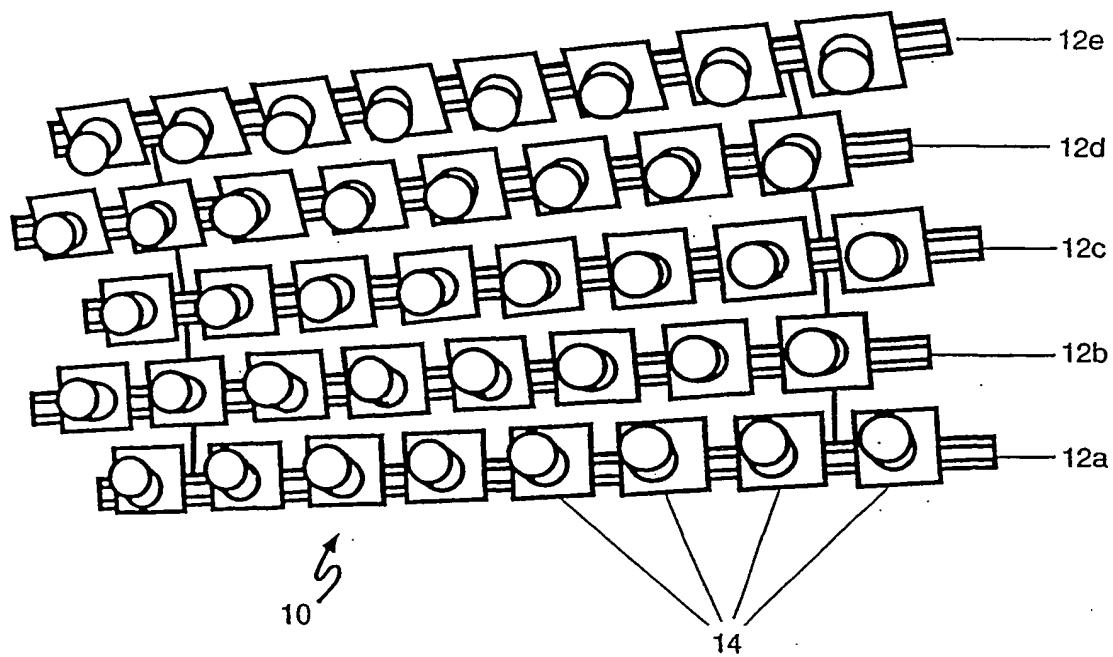


FIG. 3

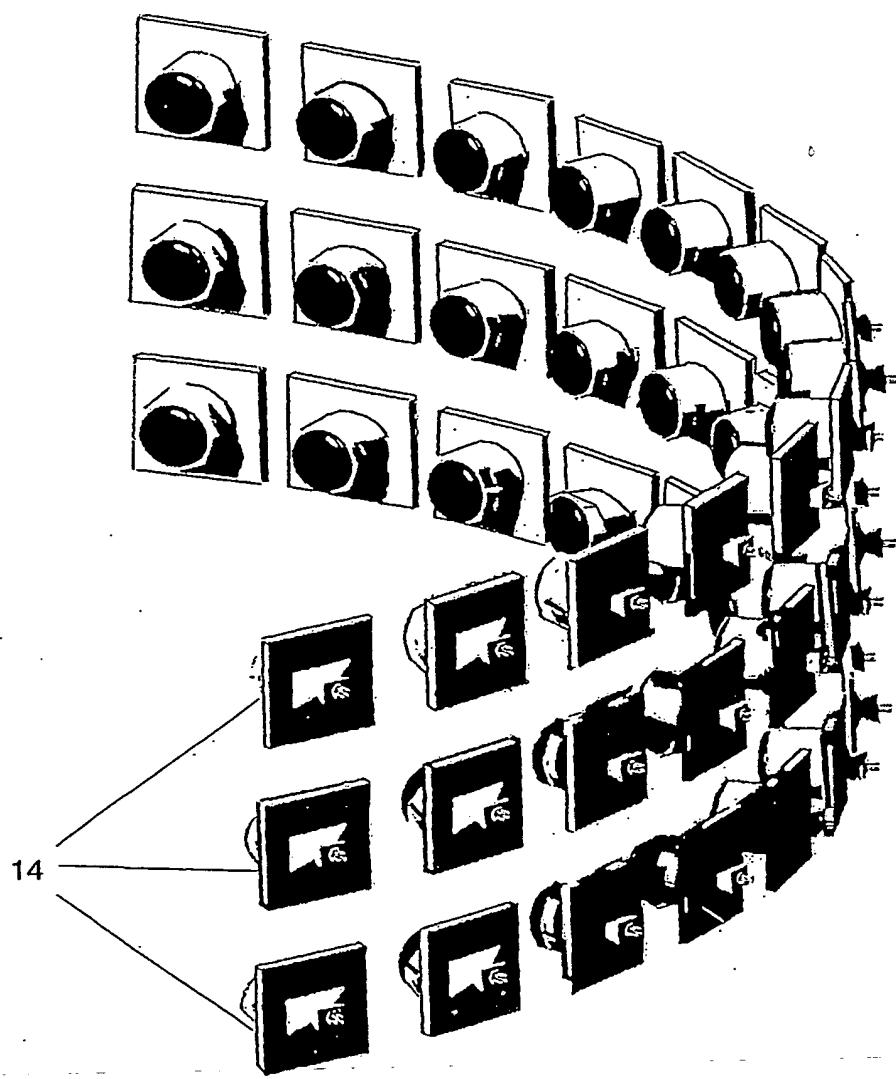


FIG. 4

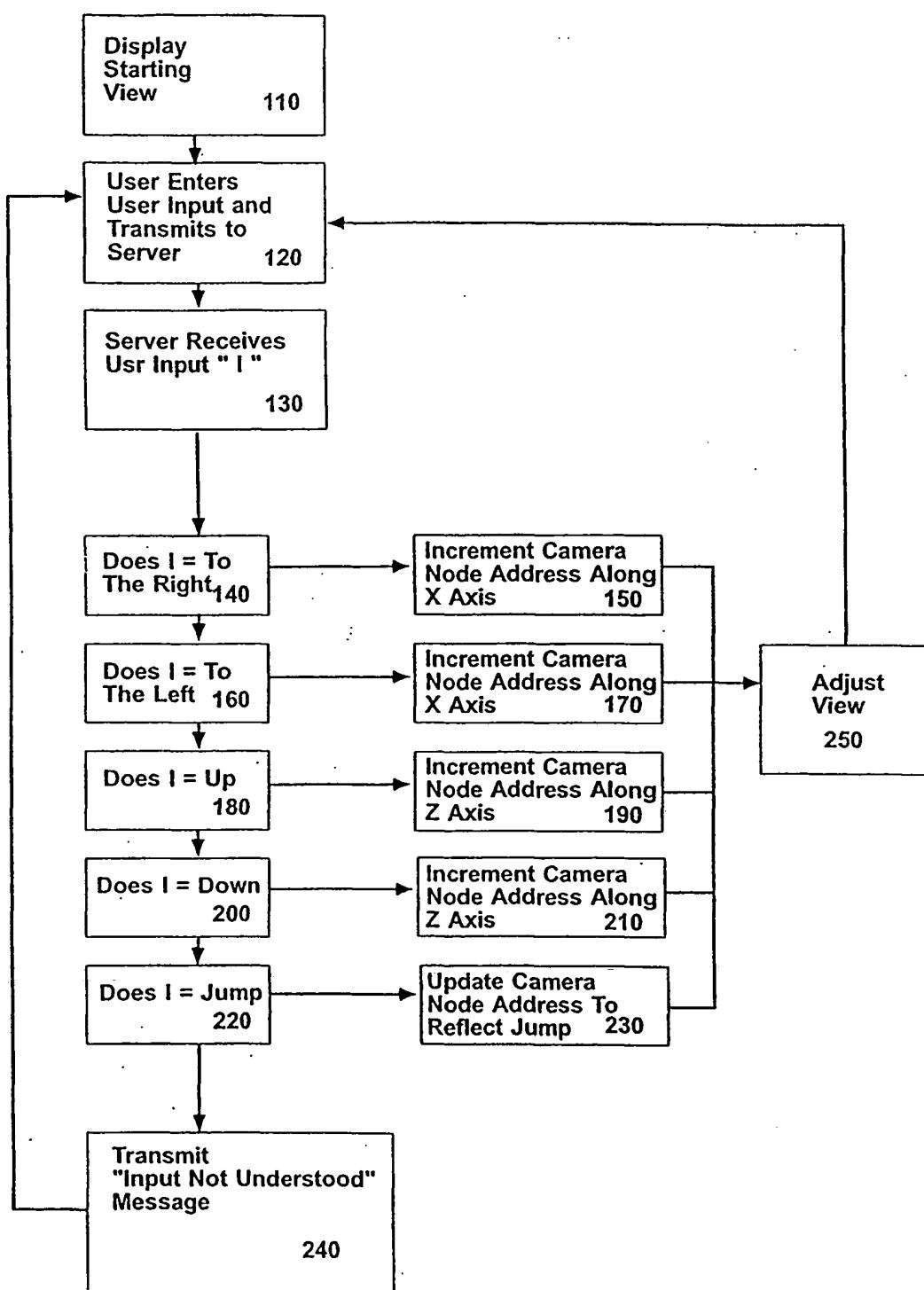


Fig. 5

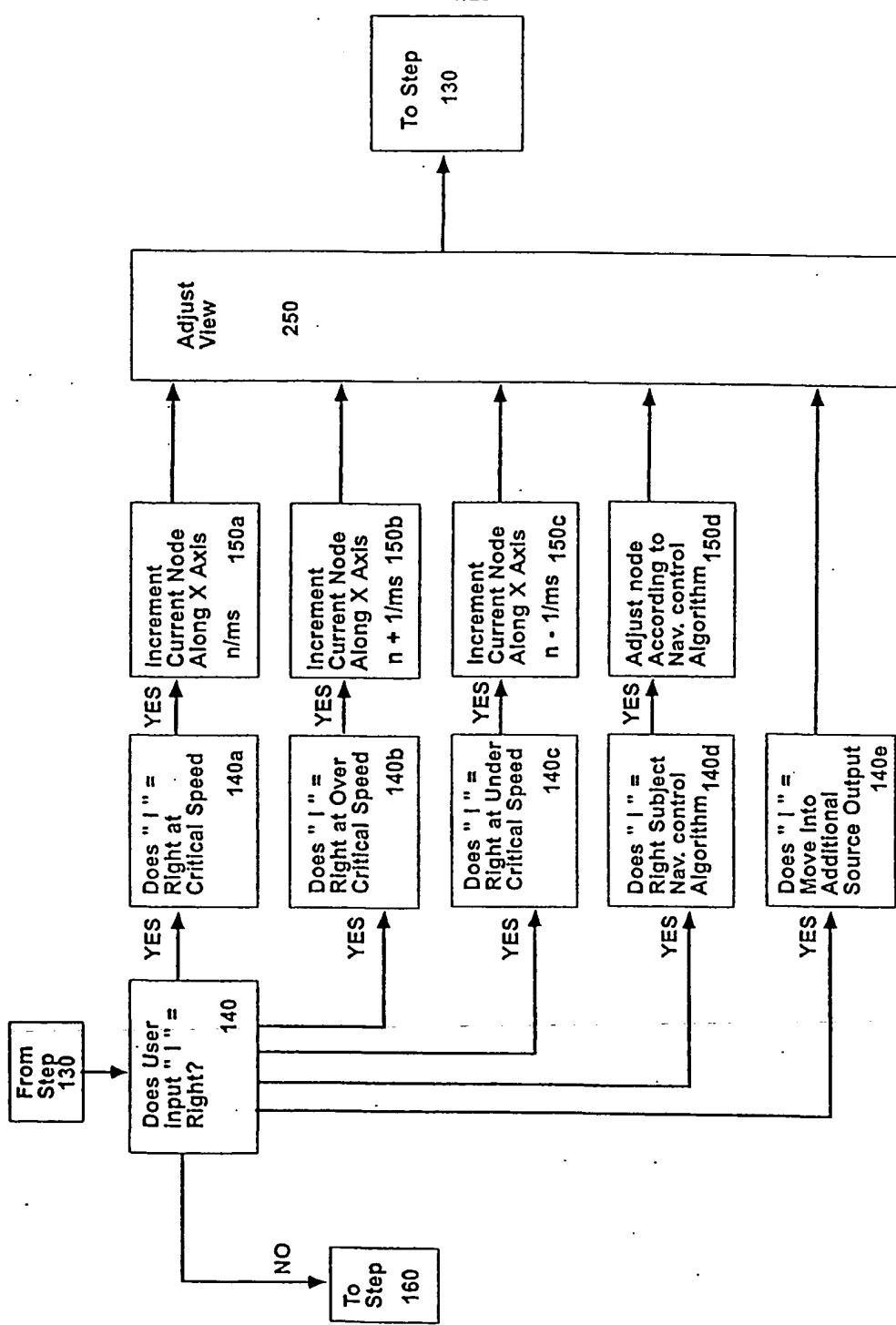


Fig. 6

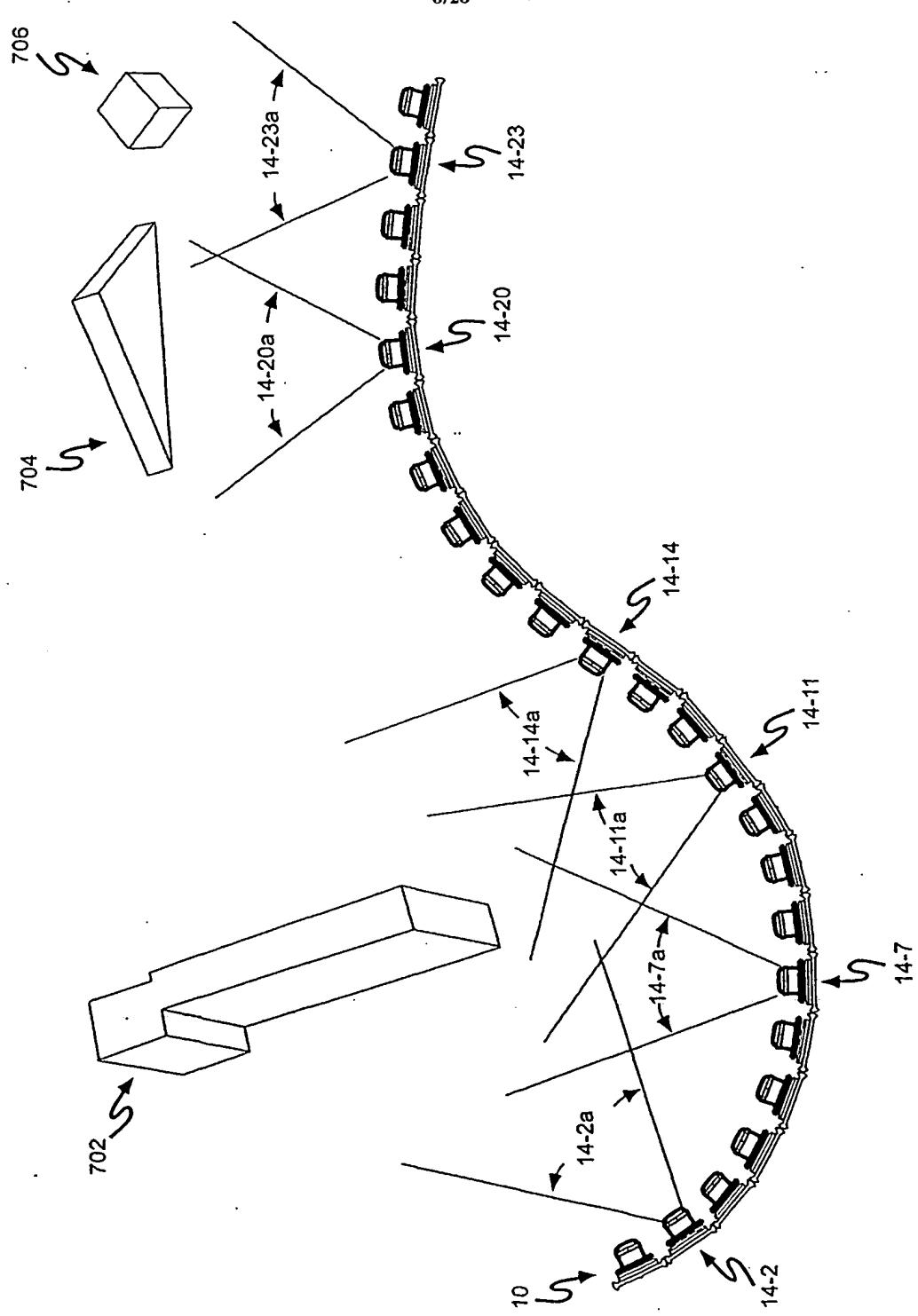


Fig.7a

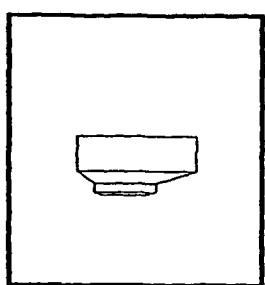


Fig. 7d

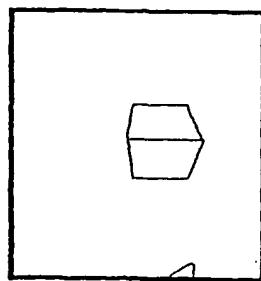


Fig. 7g

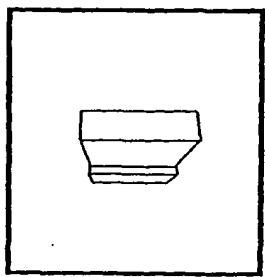


Fig. 7c

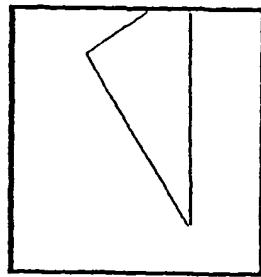


Fig. 7f

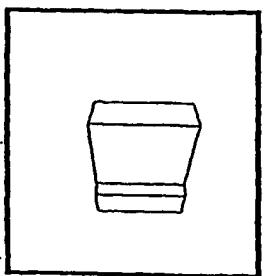


Fig. 7b

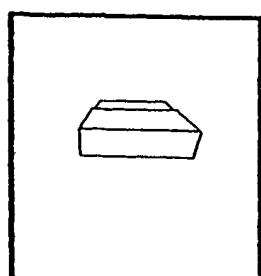


Fig. 7e

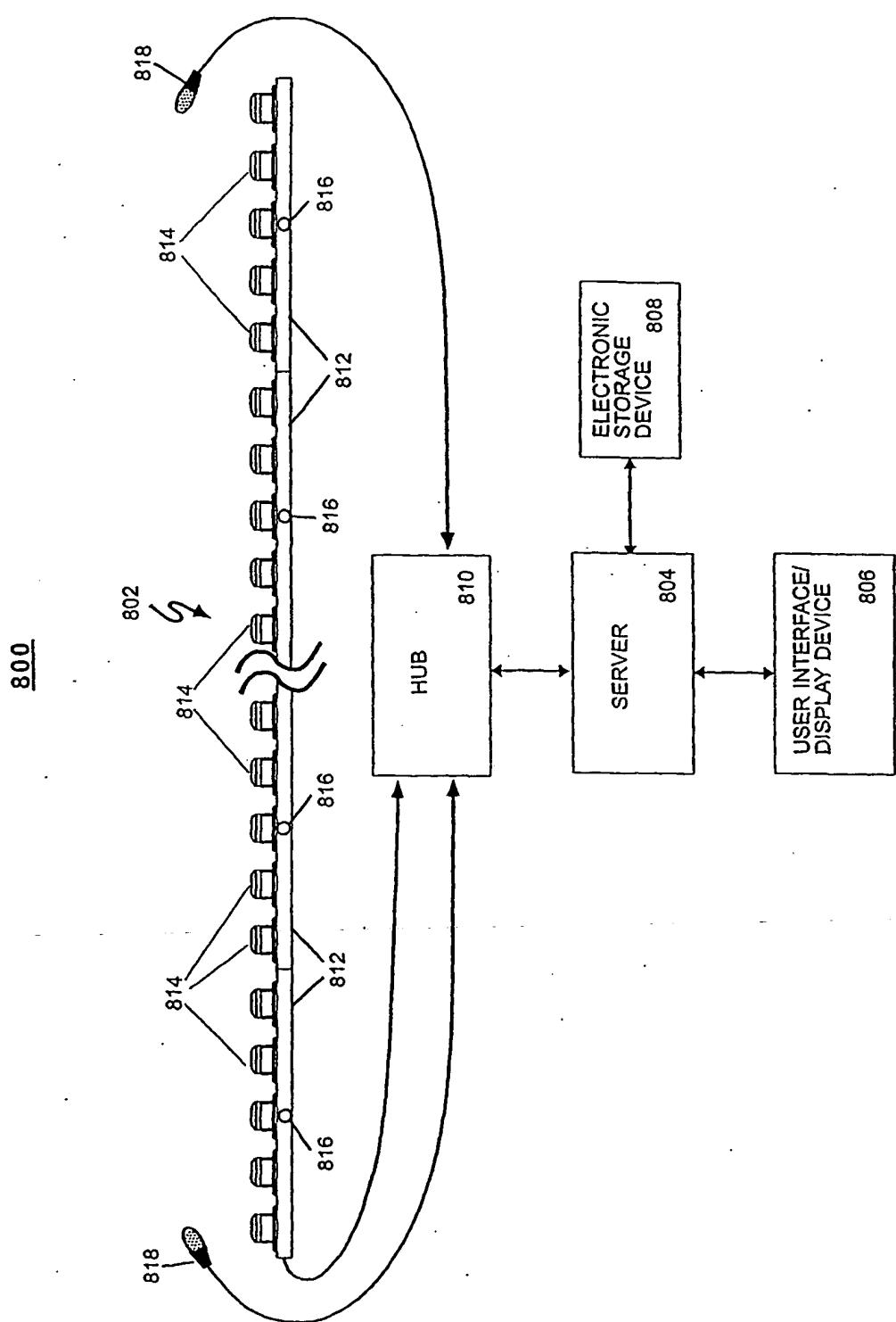


Fig. 8

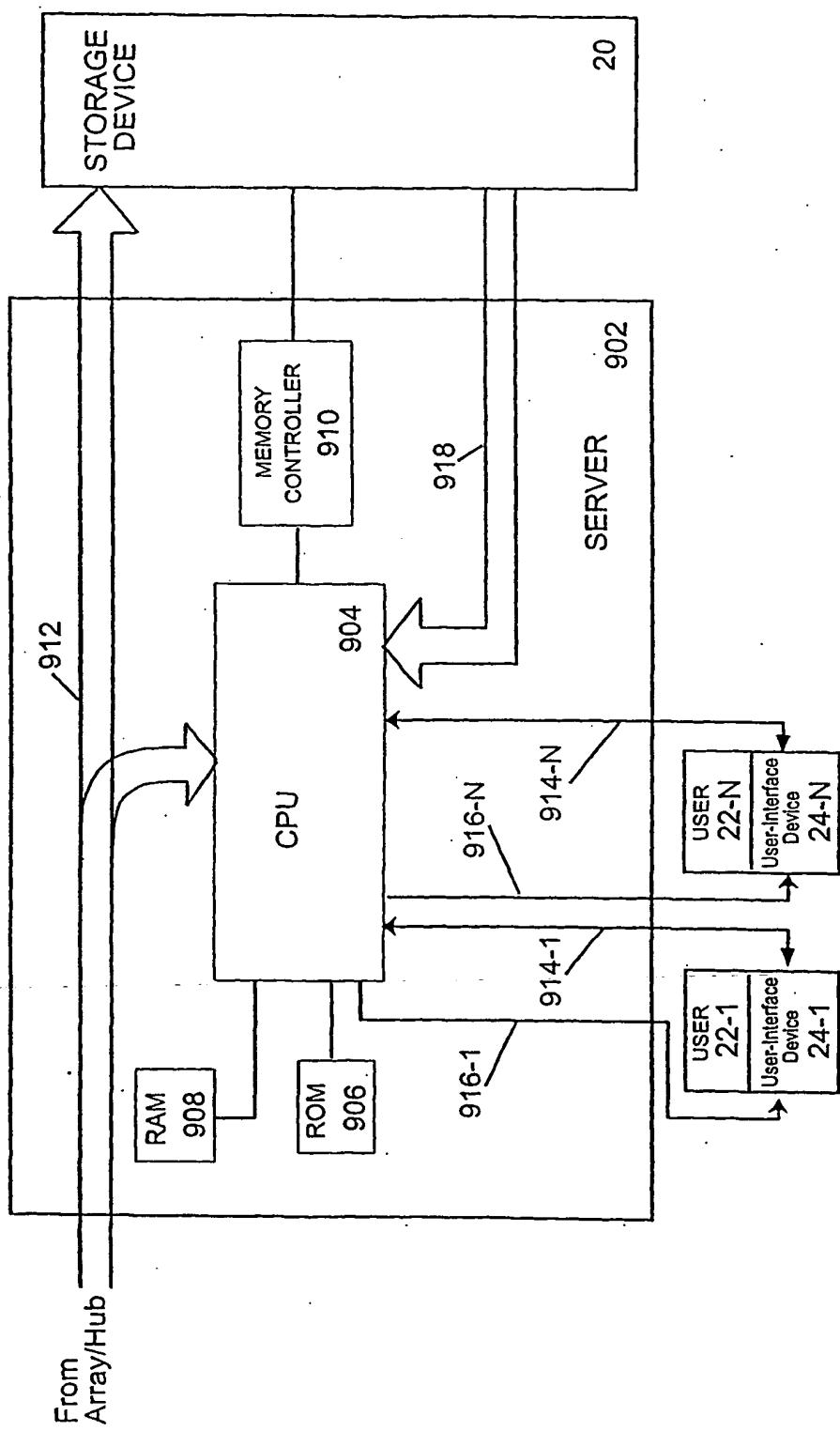


Fig. 9

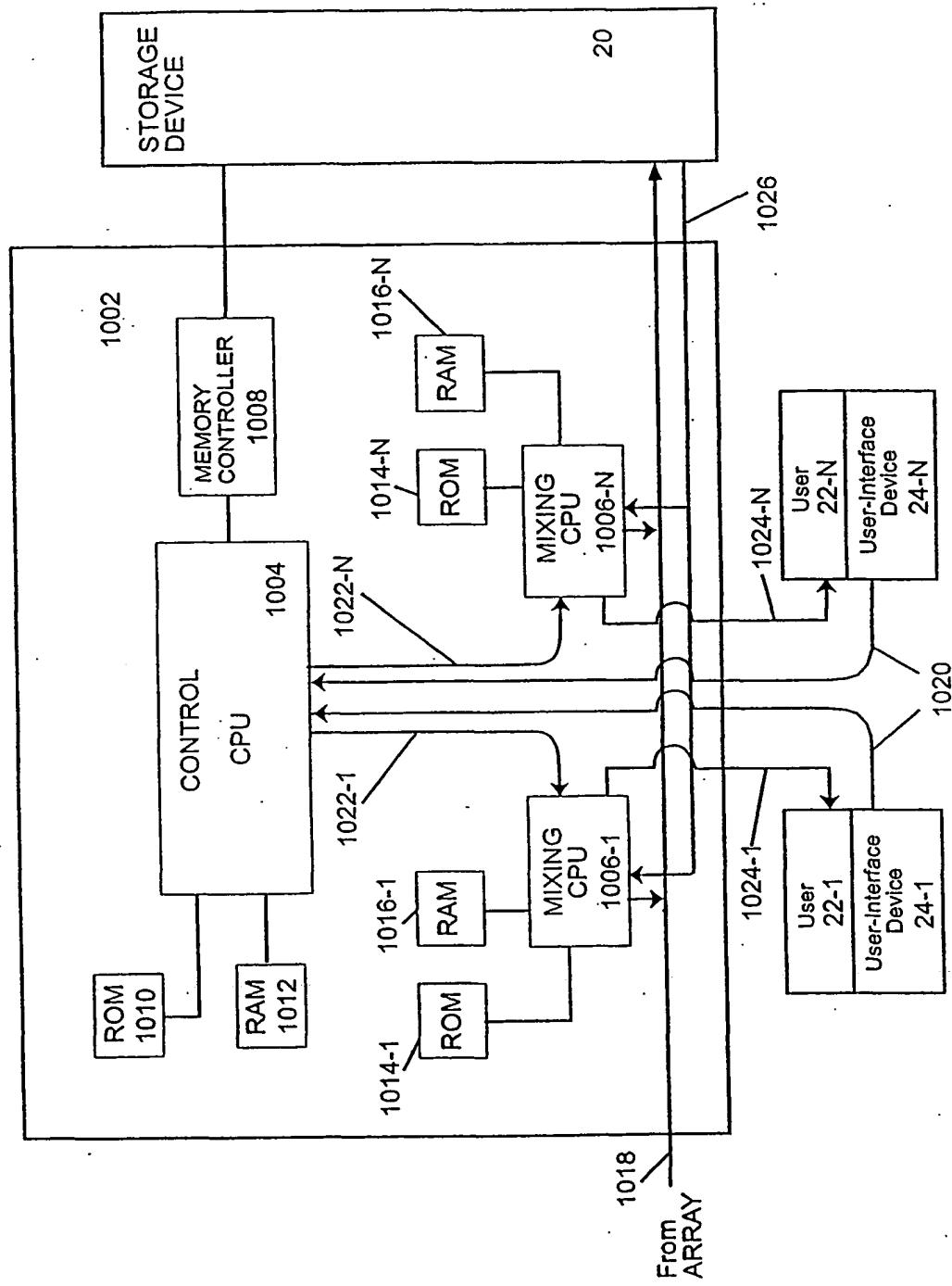


Fig. 10

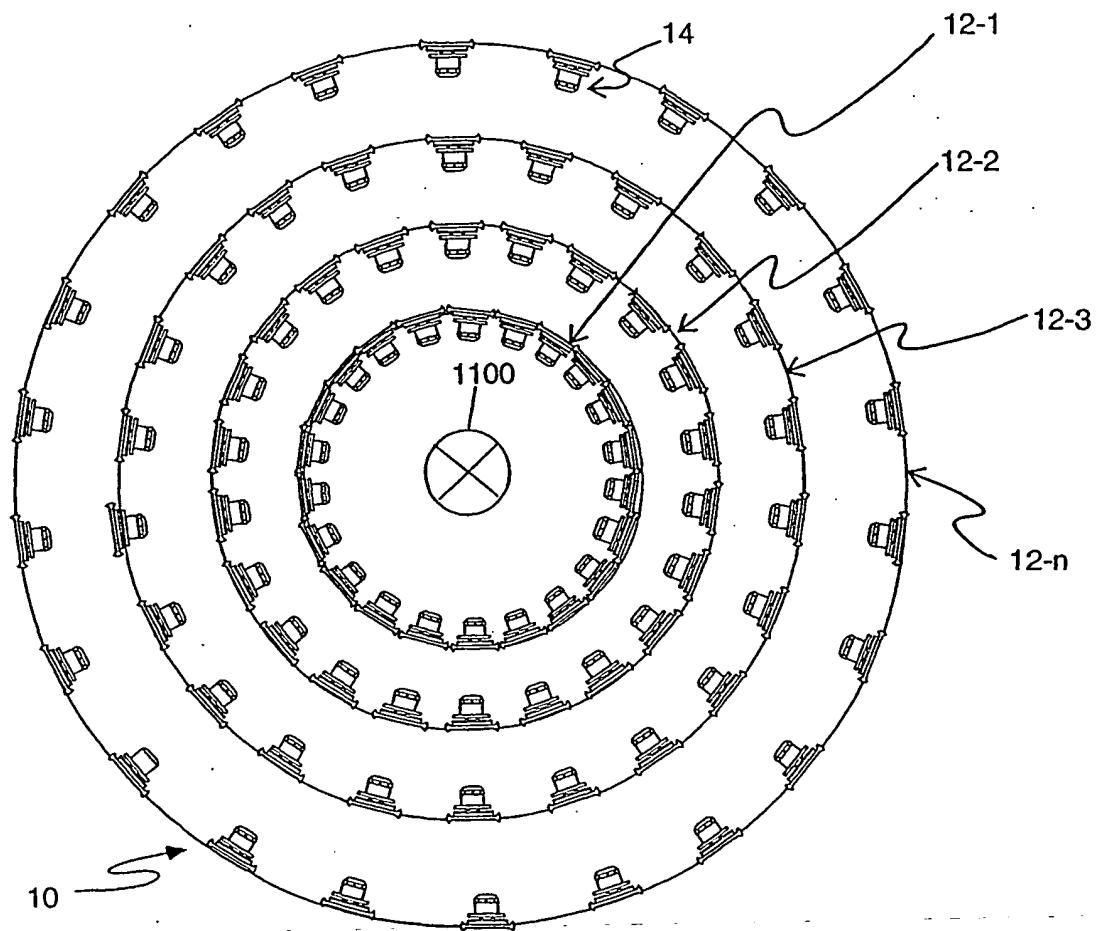


FIG. 11

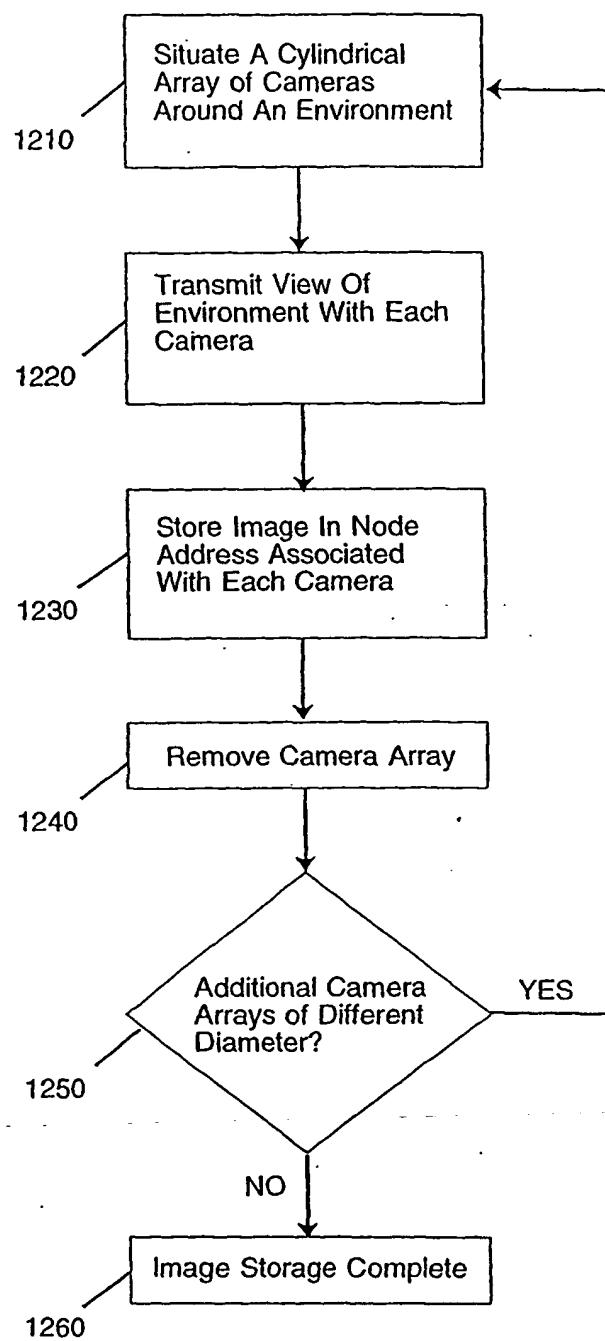


FIG. 12

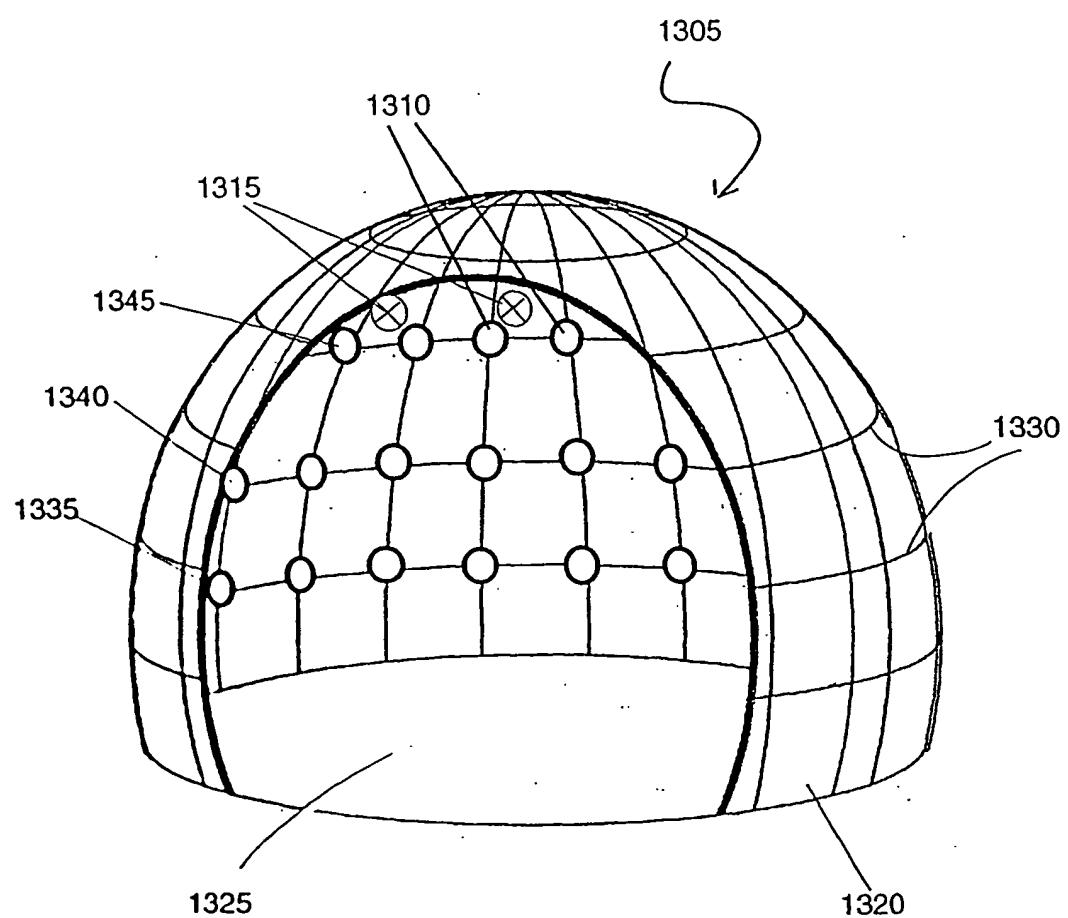


FIG. 13

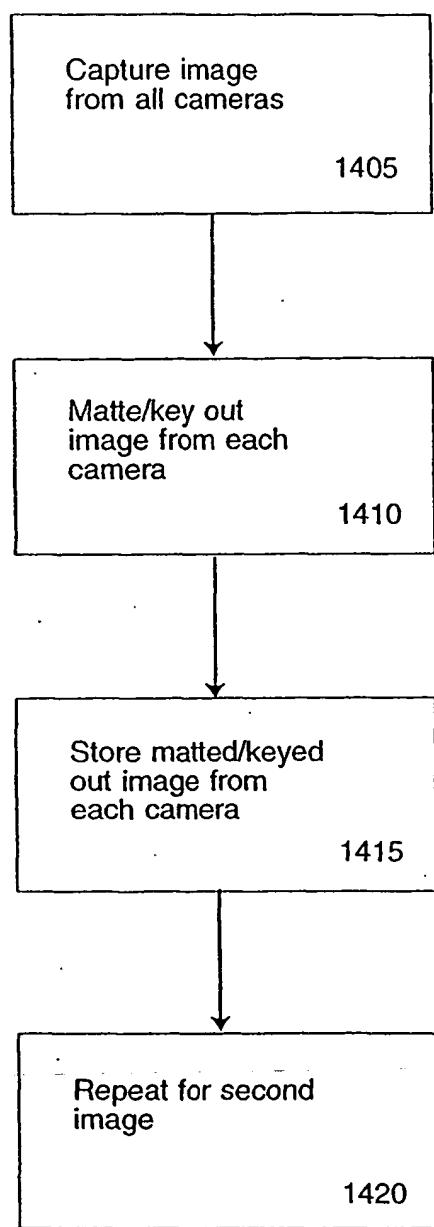


FIG. 14

**Logical Arrangement of Frames of One Image**

Camera	Address of Frame <sub>1</sub>	Address of Frame <sub>2</sub>	Address of Frame <sub>3</sub>	...
Row 1 (1)	01 01 01	01 01 02	01 01 03	
Row 1 (2)	01 02 01	01 02 02	01 02 03	
Row 1 (3)	01 03 01	01 03 02	01 03 03	
.				
.				
Row 2 (1)	02 01 01	02 01 02	02 01 03	
Row 2 (2)	02 02 01	02 02 02	02 02 03	
Row 2 (3)	02 03 01	02 03 02	02 03 03	
.				
.				
.				

**FIG. 15**

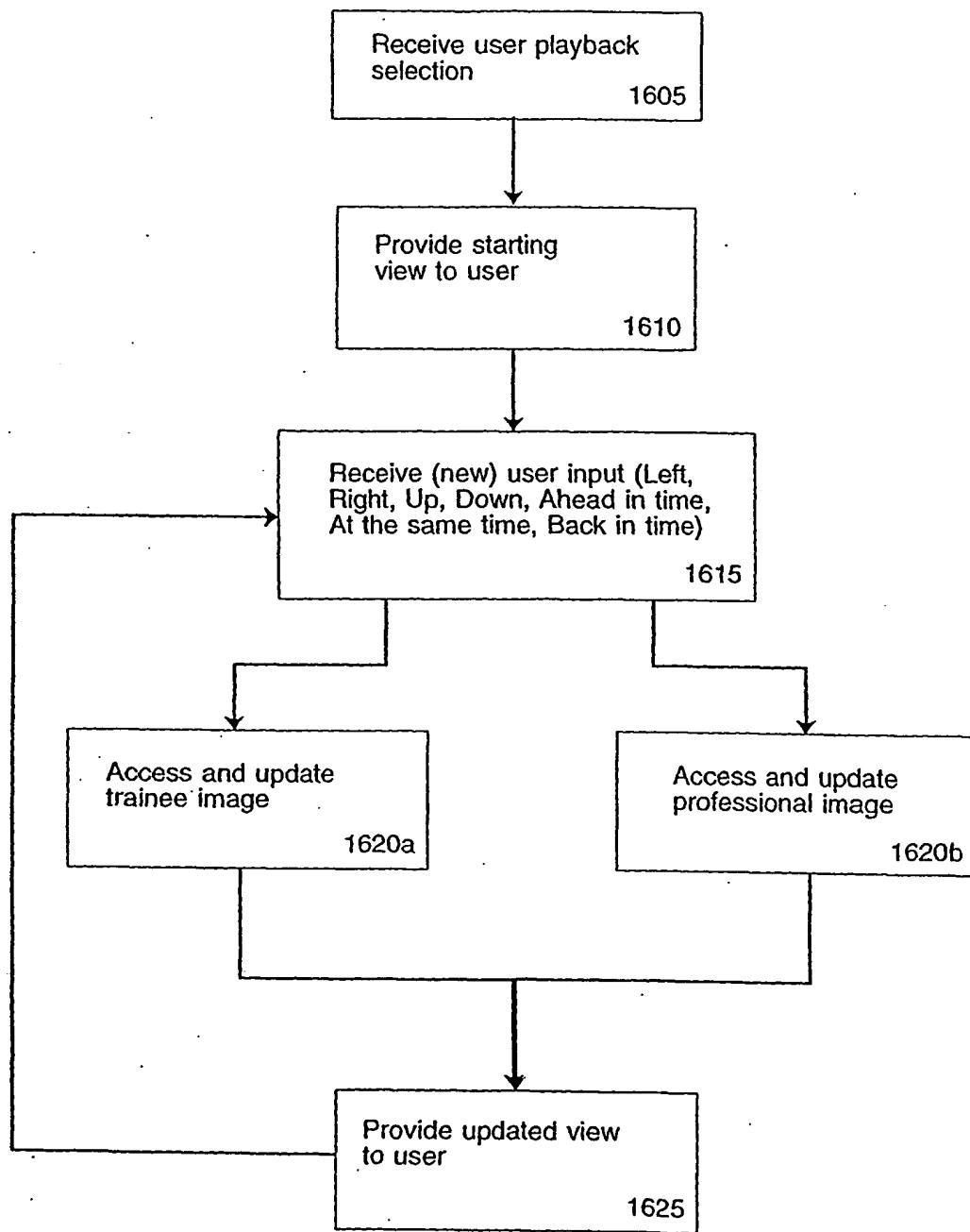


FIG. 16

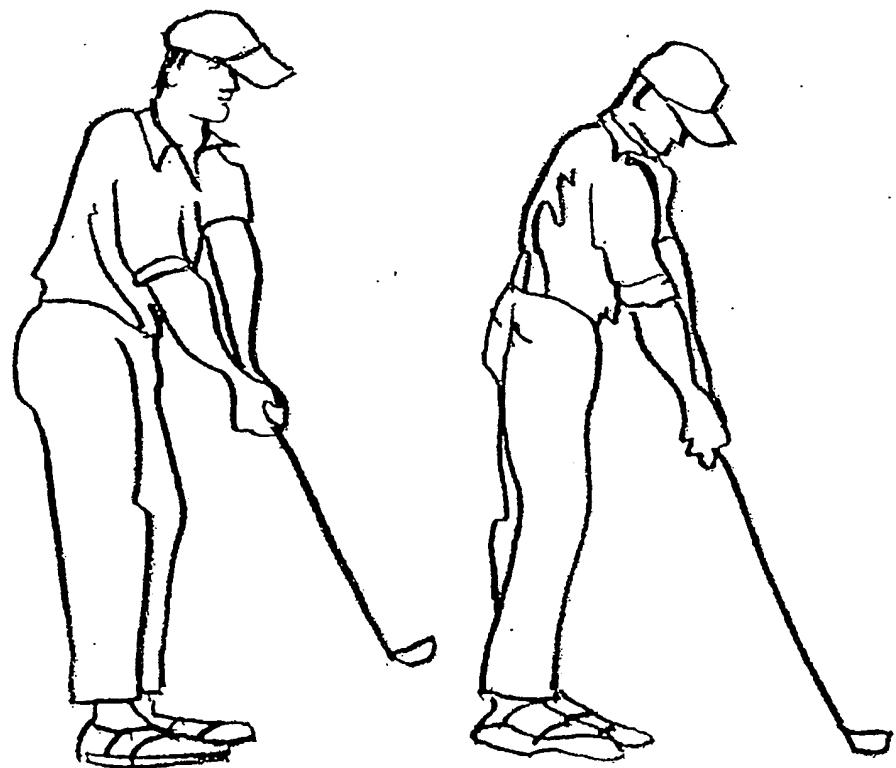


FIG. 17

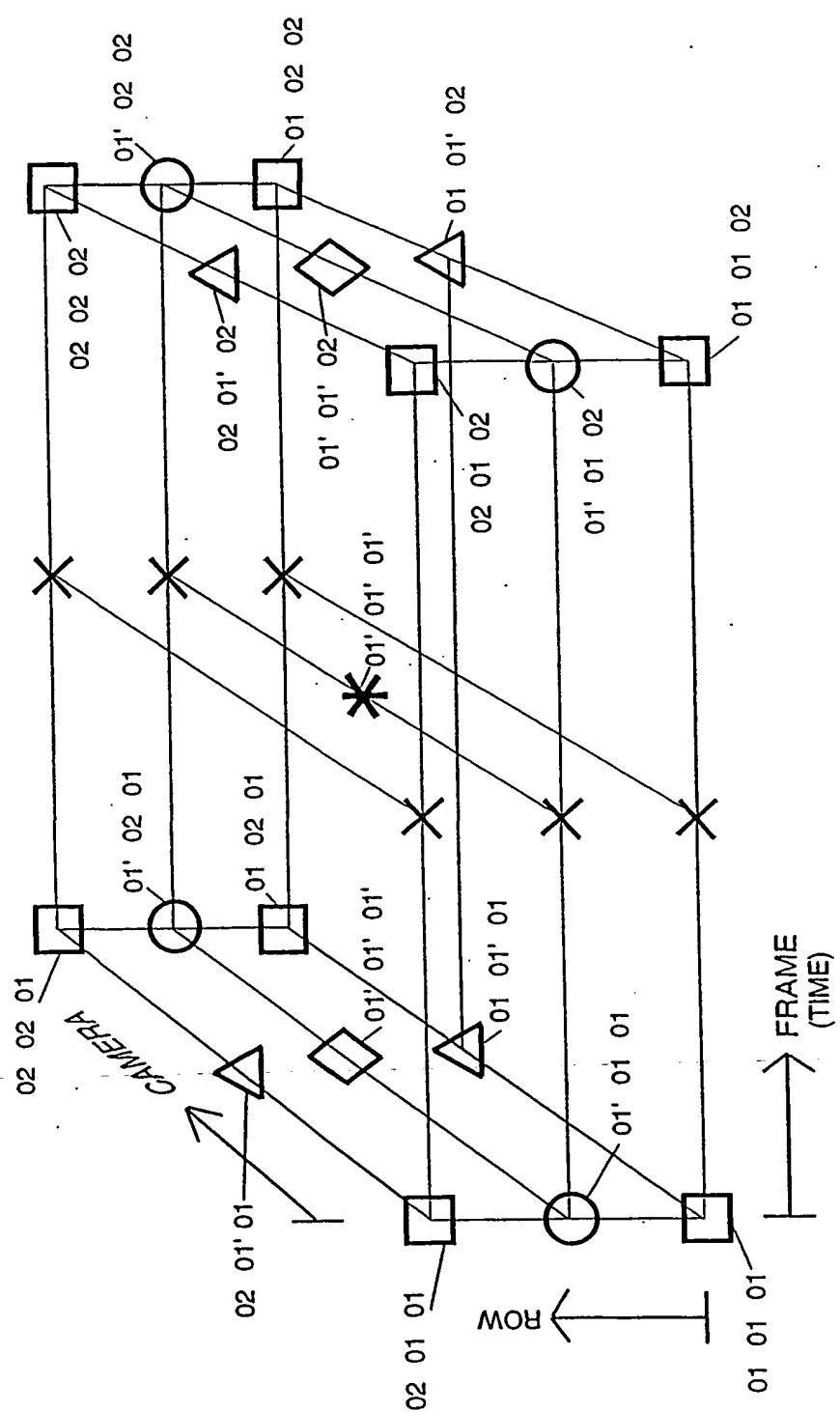


FIG. 18a

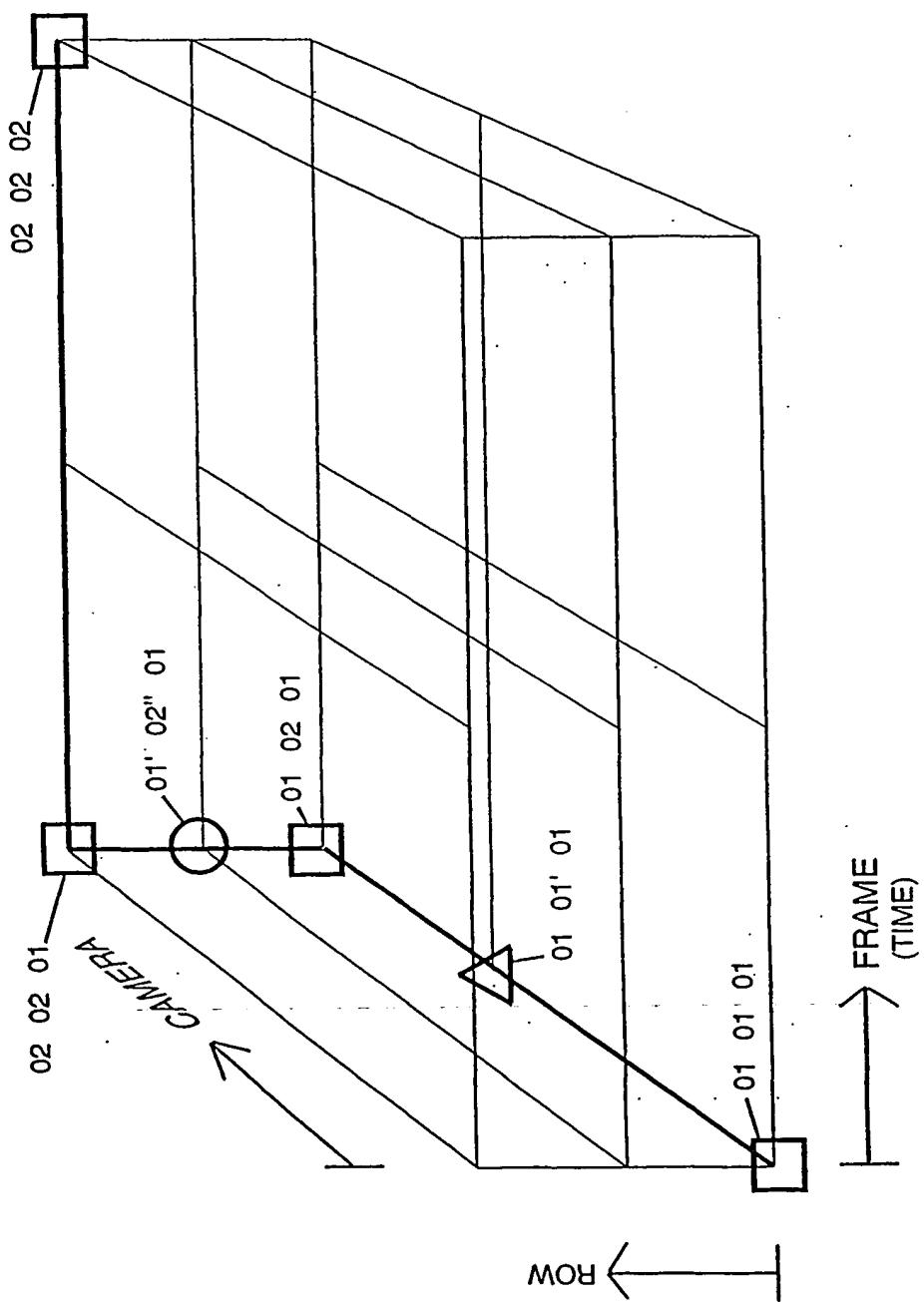


FIG. 18b

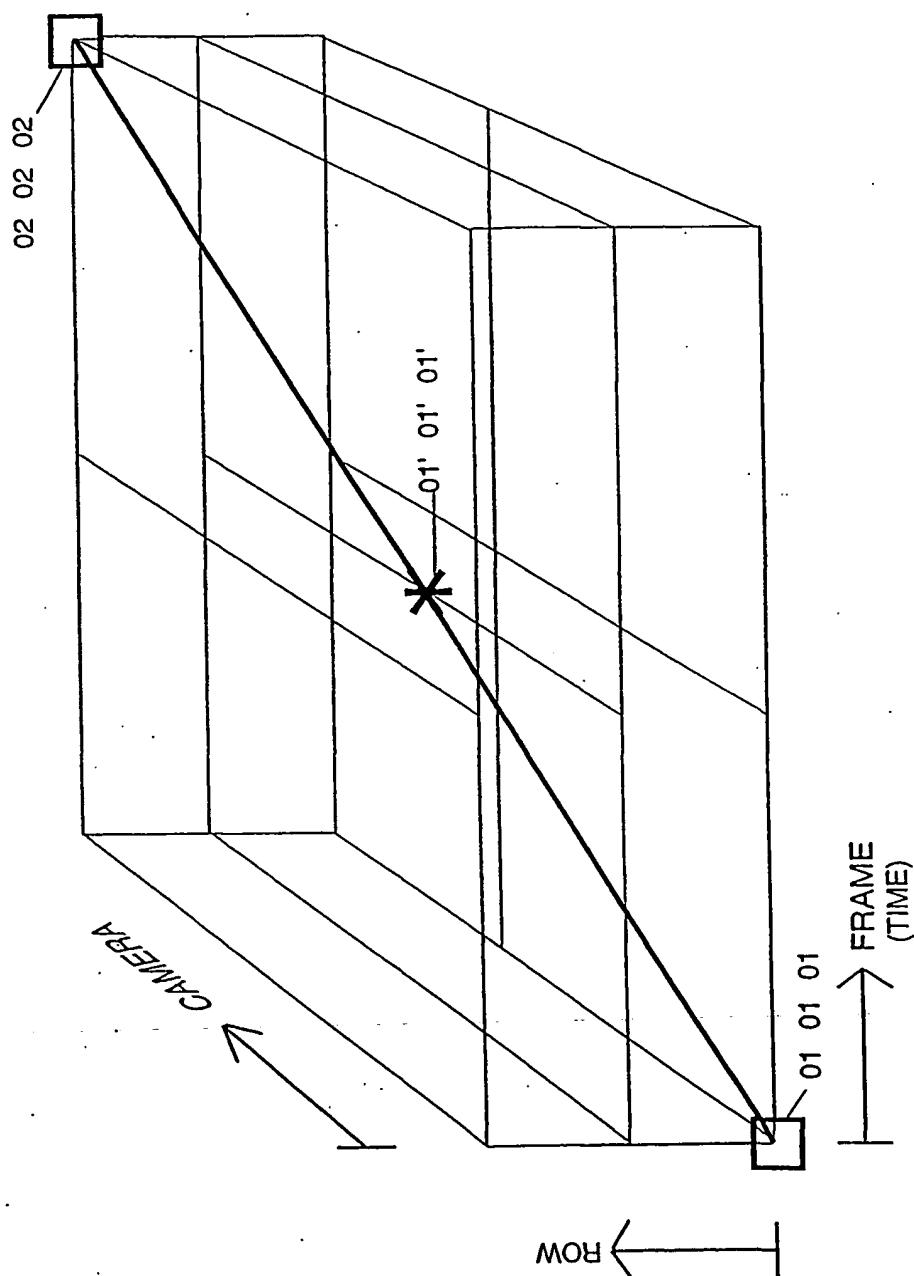


FIG. 18c

## Logical Arrangement of Frames of Two Images

## PROFESSIONAL

Camera	Frame 1	Frame 2	Frame 3	Mixed Image	Frame 4	...	Frame 60
Row 1 (1)	01 01 01	01 01 01	01 01 02	01 01 03	01 01 03	01 01 04	01 01 60
Row 1 (2)	01 02 01	01 02 01	01 02 02	01 02 03	01 02 03	01 02 04	01 02 60
Row 1 (3)	01 03 01	01 03 01	01 03 02	01 03 03	01 03 03	01 03 04	01 03 60
:							
:							

## TRAINEE

Camera	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	...	Frame 90
Row 1 (1)	01 01 01	01 01 02	01 01 03	01 01 04	01 01 05	01 01 06	01 01 90	
Row 1 (2)	01 02 01	01 02 02	01 02 03	01 02 04	01 02 05	01 02 06	01 02 90	
Row 1 (3)	01 03 01	01 03 02	01 03 03	01 03 04	01 03 05	01 03 06	01 03 90	
:								
:								

FIG. 19

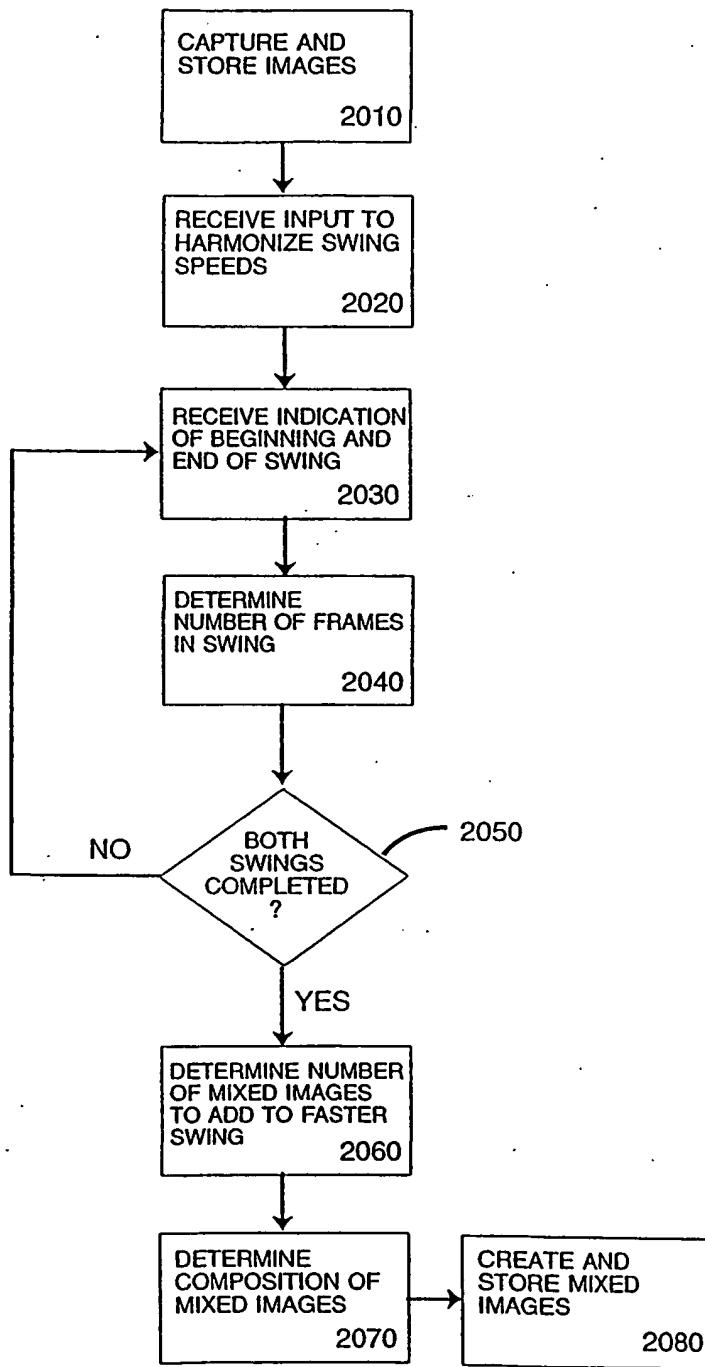


FIG. 20

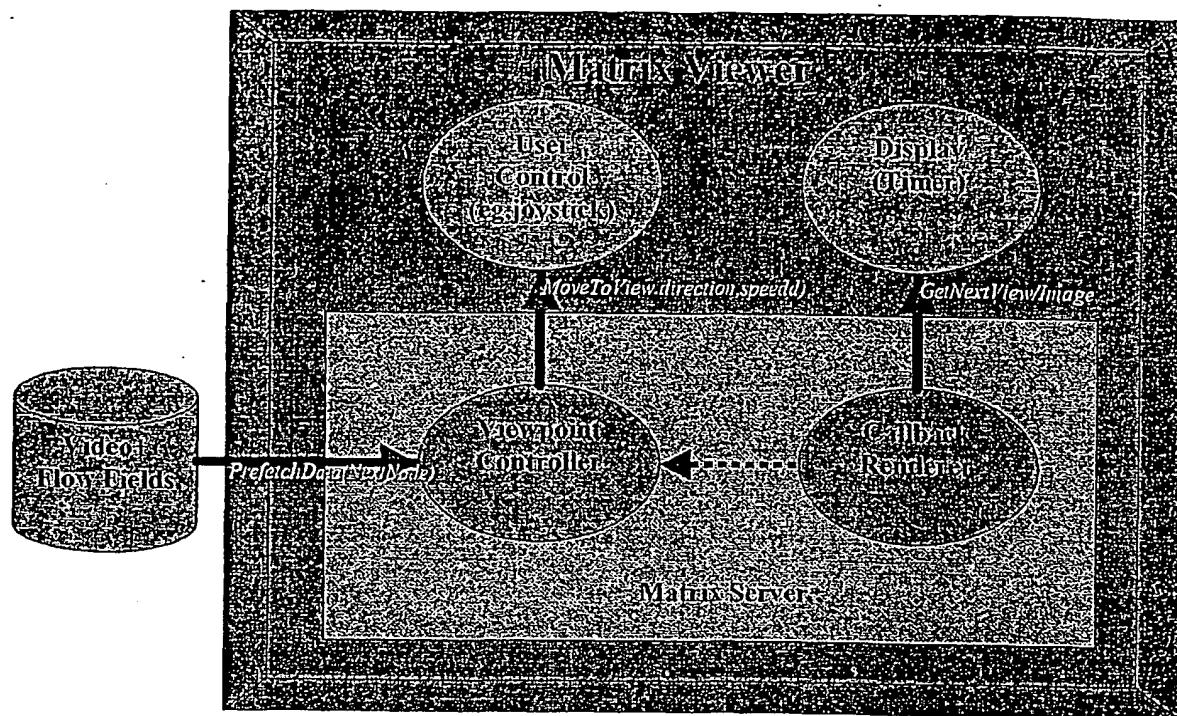


FIG. Z1

# Kewazinga

## System Configuration

FIG 22

